UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

José Eduardo Henriques da Silva

Inferência de Redes de Regulação Gênica a partir de Séries Temporais via Meta-heurísticas

Juiz de Fora
2024

José Eduardo Henriques da Silva

Inferência de Redes de Regulação Gênica a partir de Séries Temporais via Meta-heurísticas

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Modelagem Computacional. Área de concentração: Sistemas Computacionais Aplicados.

Orientador: D. Sc. Heder Soares Bernardino

Coorientadores: D. Sc. Itamar Leite de Oliveira e D. Sc. José Jerônimo Camata

Juiz de Fora

2024

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Henriques da Silva, José Eduardo.
Inferência de Redes de Regulação Gênica a partir de Séries Temporais via Meta-heurísticas / José Eduardo Henriques da Silva. -- 2019.
2024 f. : il.

Orientador: Heder Soares Bernardino
Coorientadores: Itamar Leite de Oliveira, José Jerônimo Camata
Tese (doutorado) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Modelagem Computacional, 2019.

1. Rede de Regulação Gênica. 2. Metaheurísticas. 3. Programação Genética Cartesiana. I. Soares Bernardino, Heder, orient. II. Leite de Oliveira, Itamar, coorient. III. Camata, José Jerônimo, coorient. IV. Título.

José Eduardo Henriques da Silva

Inferência de Redes de Regulação Gênica a partir de Séries Temporais via Meta-heurísticas

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Modelagem Computacional. Área de concentração: Sistemas Computacionais Aplicados.

Aprovada em 09 de abril de 2024

BANCA EXAMINADORA

Prof. D. Sc. Heder Soares Bernardino - Orientador
Universidade Federal de Juiz de Fora - UFJF

Prof. D. Sc. Itamar Leite de Oliveira - Coorientador
Universidade Federal de Juiz de Fora - UFJF

Prof. D. Sc. José Jerônimo Camata - Coorientador
Universidade Federal de Juiz de Fora - UFJF

Prof. D. Sc. Priscila Vanessa Zabala Capriles Goliatt
Universidade Federal de Juiz de Fora - UFJF

Prof. D. Sc. Alex Borges Vieira
Universidade Federal de Juiz de Fora - UFJF

Prof. D. Sc. Douglas Adriano Augusto
Fundação Oswaldo Cruz - Fiocruz

Prof. D. Sc. Ronaldo Ribeiro Goldschmidt
Instituto Militar de Engenharia - IME

Documento assinado eletronicamente por **Douglas Adriano Augusto**, **Usuário Externo**, em 10/04/2024, às 16:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Ronaldo Ribeiro Goldschmidt**, **Usuário Externo**, em 11/04/2024, às 07:31, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Heder Soares Bernardino**, **Professor(a)**, em 14/04/2024, às 21:33, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Priscila Vanessa Zabala Capriles Goliatt**, **Professor(a)**, em 22/04/2024, às 16:09, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Jose Jeronimo Camata**, **Professor(a)**, em 22/04/2024, às 20:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Itamar Leite de Oliveira**, **Professor(a)**, em 22/04/2024, às 22:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Alex Borges Vieira**, **Professor(a)**, em 23/04/2024, às 14:59, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **José Eduardo Henriques da Silva**, **Usuário Externo**, em 24/04/2024, às 15:12, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1755314** e o código CRC **F4D63CF2**.

Aos meus pais, amigos, orientador e coorientadores pelo apoio incondicional.

AGRADECIMENTOS

Agradeço ao meu orientador Heder Soares Bernardino pela disponibilidade e diversas horas de conversas e esclarecimento de dúvidas e por sua humildade em transferir-me parte de seu conhecimento e contribuir de maneira imensurável para meu crescimento.

Aos meus coorientadores Itamar e José Camata pelo auxílio imprescindível em suas áreas de especialidade.

Aos alunos de iniciação científica, Luciana e Patrick, que dedicaram-se a entender os conceitos abordados no meu doutorado e seus esforços, que resultaram em contribuições para este trabalho.

Ao apoio dos familiares, em especial minha mãe, Valéria, e meu pai, José Luiz (*in memoriam*), que sempre estimularam-me nos estudos e puderam proporcionar-me tal oportunidade.

Aos amigos que foram fundamentais para o desenvolvimento e conclusão deste trabalho. Um agradecimento especial ao meu grande amigo Evandro Bastos, sempre presente nas horas boas e ruins, pelas conversas que sempre contemplaram assuntos muito além do escopo deste trabalho e auxiliaram-me, não só na trajetória acadêmica, mas também no desenvolvimento pessoal.

Aos amigos de UNIFAA, Reinaldo, Mariana e Galba pela confiança, companheirismo, e troca de experiências, além das inúmeras conversas.

A todos que diretamente ou indiretamente contribuíram para a concretização deste trabalho.

Agradeço também à Universidade Federal de Juiz de Fora e ao programa de Pós-Graduação em Modelagem Computacional pela infraestrutura e fomento para o desenvolvimento deste trabalho, bem como à CAPES.

“Ninguém é tão grande que não possa aprender, nem tão pequeno que não possa ensinar.”

Esopo

RESUMO

A inferência de redes de regulação gênica (GRNs - do inglês *Gene Regulatory Networks*) é um problema difícil e importante, com desafios amplamente endereçados na área denominada Biologia Sistêmica. Suas aplicações incluem biotecnologia e saúde, auxiliando no desenvolvimento de fármacos, uma vez que a compreensão de padrões nas interações gênicas pode levar a descobertas importantes relacionadas a doenças nos organismos. O sequencimento de RNA de célula única (scRNA-Seq - do inglês *single-cell RNA Sequencing*) proveu uma resolução sem precedentes para o campo da transcriptômica. Experimentos que utilizam scRNA-Seq são atrativos para a inferência de GRNs devido à geração de milhares de medidas independentes e à possibilidade de se obter uma visão pseudotemporal mais precisa da dinâmica da expressão gênica. Entretanto, nem todos os genes são expressos o tempo todo. A seleção de conjuntos de genes que modelam o fenômeno biológico desejado também constitui um desafio para a inferência de GRNs. As redes Booleanas e as modeladas por meio de sistemas de equações diferenciais ordinárias (EDOs) são comumente utilizadas para representar as GRNs. Contudo, não existe método padrão para discretização dos dados que são fornecidos às redes Booleanas. Redes Booleanas podem ser modeladas na forma de circuitos digitais. Dentre as técnicas de computação evolucionista, Programação Genética Cartesiana (CGP - do inglês *Cartesian Genetic Programming*) é apontada como a técnica mais eficiente para a evolução e otimização de circuitos lógicos combinacionais. Entretanto, técnicas de computação evolucionista não aparecem dentre os algoritmos destacados como estado da arte para a reconstrução de GRNs, motivado principalmente por problemas de escalabilidade. Além disso, o desconhecimento das redes *ground-truth* e não padronização da forma de atribuir qualidade à uma rede inferida aumentam o desafio ao resolver o problema. Neste trabalho propõe-se um *framework* que utiliza CGP para a inferência de GRNs Booleanas e a obtenção de um modelo contínuo a partir de dados na forma de séries temporais. Cada etapa do *framework* proposto é explorada, abrangendo (i) o pré-processamento dos dados de expressão gênica, (ii) a seleção de subconjuntos de genes via técnicas de agrupamento como forma de direcionar o processo de busca, (iii) as maneiras pelas quais os dados devem ser discretizados a fim de se obter um modelo Booleano, (iv) o comportamento dos operadores de variação genética na CGP, (v) a forma pela qual um modelo Booleano pode ser convertido em um sistema de EDOs e (vi) a determinação dos coeficientes numéricos deste sistema de EDOs via Estratégias Evolutivas. Propõe-se, também, um novo procedimento para discretização de dados de expressão gênica na forma de séries temporais. Por fim, uma revisão do processo metodológico adotado no contexto de inferência de redes de regulação gênica a partir de dados scRNA-Seq, abrangendo as características intrínsecas à tecnologia de sequenciamento, a seleção de genes de interesse, os *motifs* de rede, as redes de referência e as métricas e forma de avaliar as redes inferidas é apresentada.

Como resultado, propõe-se um novo processo metodológico. Todas as propostas são avaliadas em problemas *benchmark*, que consideram dados sintéticos e reais obtidos por meio de *microarrays* e scRNA-Seq, dados oriundos de simulação estocástica, além de dados de organismos amplamente conhecidos e explorados na literatura, como *Saccharomyces cerevisiae* e *Escherichia coli*, e dados da competição DREAM4. Os resultados mostram que as propostas são superiores ou competitivas com os métodos estado da arte para a inferência de GRNs e fornecem uma solução interpretável que pode auxiliar os especialistas do domínio no campo de Biologia Sistêmica. Além disso, o processo metodológico proposto torna mais justa a comparação de diferentes algoritmos de inferência de GRNs.
Palavras-chave: Rede de Regulação Gênica. Metaheurísticas. Programação Genética Cartesiana.

ABSTRACT

The inference of gene regulatory networks (GRNs) is a difficult and important problem, with challenges largely addressed in the area called Systems Biology. Its applications include biotechnology and health, assisting in the development of drugs, since understanding patterns in gene interactions can lead to important discoveries related to diseases in organisms. Single-cell RNA sequencing (scRNA-Seq) has provided unprecedented resolution to the field of transcriptomics. Experiments using scRNA-Seq are attractive for the inference of GRNs due to the generation of thousands of independent measurements and the possibility of obtaining a more accurate pseudotemporal view of the dynamics of gene expression. However, not all genes are expressed all the time. The selection of gene subsets that model the desired biological phenomenon also constitutes a challenge for the inference of GRNs. Boolean networks and those modeled through systems of ordinary differential equations (ODEs) are commonly used to represent GRNs. Nevertheless, there is no standard method for discretizing the data that is provided to Boolean networks. Boolean networks can be modeled in the form of digital circuits. Among evolutionary computing techniques, Cartesian Genetic Programming (CGP) is considered the most efficient technique for the evolution and optimization of combinational logic circuits. However, evolutionary computing techniques do not appear among the algorithms highlighted as state of the art for reconstructing GRNs, mainly motivated by scalability problems. Furthermore, the lack of knowledge about *ground-truth* networks and the non-standardization of the way to attribute quality to an inferred network increase the challenge when solving the problem. In this work, we propose a *framework* that uses CGP to infer Boolean GRNs and obtain a continuous model from data in the form of time series. Each step of the proposed *framework* is explored, covering the pre-processing of gene expression data, the selection of subsets of genes via clustering techniques as a way of directing the search process, the ways in which the data should be discretized in order to obtain a Boolean model, the behavior of the genetic variation operators in the CGP, the way in which a Boolean model can be converted into a system of ODEs and the determination of the numerical coefficients of this system of ODEs via Evolutionary Strategies. A new procedure for discretizing gene expression data in the form of time series is also proposed. Finally, a review of the methodological process adopted in the context of inferring gene regulation networks from scRNA-Seq data, covering the intrinsic characteristics of sequencing technology, the selection of genes of interest, the network *motifs* , the reference networks and the metrics and way to evaluate the inferred networks are presented. As a result, a new methodological process is proposed. All proposals are evaluated in *benchmark* problems, which consider synthetic and real data obtained through *microarrays* and scRNA-Seq, data from stochastic simulation, in addition to data from organisms widely known and explored in the literature , such as *Saccharomyces cerevisiae*

and *Escherichia coli*, and data from the DREAM4 competition. The results show that the proposals are superior or competitive with state-of-the-art methods for the inference of GRNs and provide an interpretable solution that can assist domain experts in the field of Systemic Biology. Furthermore, the proposed methodological process makes the comparison of different GRN inference algorithms fairer.
Keywords: Gene Regulatory Network. Metaheuristics. Cartesian Genetic Programming.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

A	Adenina
ACC	Accuracy
AE	Algoritmo Evolutivo
AUPRC	Área sob a curva *precision-recall*
AUROC	Área sob a curva *ROC*
BACC	Balanced Accuracy
BKM	Bikmeans
bps	Pares de Base (do inglês - *base pairs*)
C	Citosina
cDNA	DNA complementar (do inglês - *complementary DNA*)
CGP	Programação Genética Cartesiana (do inglês - *Cartesian Genetic Programming*)
ChIP-Seq	*cell-type specific ChIP-Seq*
CLC	Circuito Lógico Combinacional (do inglês - *Combinational Logic Circuit*)
DAG	Grafo Direcionado Acíclico (do inglês - *Directed Acyclic Graph*)
DBI	*Davies-Bouldin Index*
DE	*Differential Expression*
DNA	Ácido Desoxirribonucleico (do inglês - *Deoxyribonucleic Acid*)
DP	*Differential Proportion*
DREAM4	DREAM4 - *In Silico Network Challenge*
DSSPD	*Distribution and Successive Spline Points Discretization*
dsRNA	RNA de dupla-fita
DV	*Differential Variability*)
EDO	Equações Diferenciais Ordinárias
EFD	*Equal Frequency Discretization*
EP	*Early Precision*
EPR	*Early Precision Ratio*
ES	Estratégias Evolutivas (do inglês - *Evolutionary Strategies*)
EWD	*Equal Width Discretization*
pdBf	Função Booleana parcialmente definida (do inglês - *partially defined Boolean function*)
FDR	*False Discovery Rate*
FN	*False Negative*
FNR	*False Negative Rate*
FOR	*Omission Rate*
FP	*False Positive*
FPR	*False Positive Rate*
G	Guanina
GA	*Genetic Algorithm*
GEDPROTOOLS	*Gene Expression Data Pre-Processing Tool*

GP	Programação Genética (do inglês - *Genetic Programming*)
GPU	*Graphics Processing Unit*
GRN	Rede de Regulação Gênica (do inglês - *Gene Regulatory Network*)
GSD	*Gonadal Sex Determination*
HD	*Hamming Distance*
HSC	*Hematopoietic Stem Cell*
HSPC	*Hematopoietic Stem and Progenitor cell*
HVG	*Highly Variable Genes*
IVT	Transcrição *In Vitro* (do inglês - *in vitro Transcription*)
iPSCs	*induced pluripotent stem cells*
IRMA	*In vivo Reverse-engineering and Modeling Assessment*
iRNA	interferência de RNA
lb	*levels-back*
mCAD	*Mammalian Cortical Area Development*
MCC	*Matthews Correlation Coefficient*
miRNA	micro RNA
mRNA	RNA mensageiro
NGD	Deriva Gênica Neutra (do inglês *Neutral Genetic Drift*)
NonSpecific	*NonSpecific ChIP-Seq*
NPV	*Negative Predictive Value*
PCR	Reação em Cadeia Polimerase (do inglês - *Polymerase Chain Reaction*)
piRNA	*piwi-interaction* RNA
PLA	*Programmable Logic Array*
PM	Mutação Pontual (do inglês - *Point Mutation*)
PP	*Performance Profile*
PPV	*Positive Predictive Value*
PSO	*Particle Swarm Optimization*
Q1	Primeiro Quartil
Q3	Terceiro Quartil
RNA	Ácido Ribonucleico (do inglês - *Ribonucleic Acid*)
RNAi	Interferência por RNA
RNA-Seq	*RNA-Sequencing*
ROC	*Receiver Operation Characteristic*
RT	Transcrição Reversa (do inglês - *Reverse Transcription*)
SAM	*Single Active Mutation*
scRNA-Seq	*Single-Cell RNA-Sequencing*
siRNA	*small interfering RNA*
SOMO	*Semantically-Oriented Mutation Operator*
SSE	*Sum of Squared Error*
STD	desvio padrão
T	Timina
TF	Fator de Transcrição (do inglês - *Transcription Factor*)

TN	*True Negative*
TNR	*True Negative Rate*
TP	*True Positive*
TPR	*True Positive Rate*
TSD	Diagrama de Transição de Estados (do inglês *Transition State Diagram*)
TSD	*Transitional State Discrimination*
TSS	Local de Início da Transcrição (do inglês - *Transcription Start Site*)
U	Uracila
UMI	*Unique Molecular Identifier*
VSC	*Ventral Spinal Cord Development*
WCSS	*Within-Cluster Sum of Squares*

LISTA DE SÍMBOLOS

$\forall$	Para todo
$\in$	Pertence
n_c	número de colunas (CGP)
n_r	número de linhas (CGP)
lb	*levels-back* (CGP)
L_n	número de nós (CGP)
Γ	conjunto de funções (CGP)
n_{eval}	número de avaliações da função objetivo (CGP)
N_{esp}	número de espécies (CGP)
Σ	alfabeto de discretização
μ	número de progenitores (ES)
λ	número de descendências (ES)
Θ	operador ternário (SOMO)
$\bigodot$	operador de redução (SOMO)
p_q	quantidade de nós inativos mutados (SOMO)
c	nó aleatório selecionado (SOMO)
μ_{var}	*threshold* de variância (DSSPD)
δ	*threshold* para discretização
a'_{ij}	nível de expressão gênica da linha i, coluna j
a_{ij}	estado discreto para o nível de expressão gênica da linha i, coluna j
B	função de atualização discreta
$\overline{B}$	função de atualização contínua genérica
$\overline{\dot{x_i}}$	comportamento temporal do modelo contínuo
$\overline{B_i}^H$	função de atualização contínua do tipo Hill

SUMÁRIO

1 INTRODUÇÃO

A Biologia Sistêmica é um campo interdisciplinar que envolve conhecimentos de Biologia, Química, Matemática, Física, entre outras, cujo estudo concentra-se na interação entre os componentes de um sistema biológico e como estas interações geram comportamentos e funções dentro de um sistema (WANG; SAADATPOUR; ALBERT, 2012; PALSSON, 2015; KLIPP *et al.*, 2016). O estudo de redes celulares (regulação gênica, proteína e metabólicas) envolvem interações coordenadas de milhares de moléculas e são alvo central de estudo no campo de Biologia Sistêmica (SAURO, 2014; PETRATOU *et al.*, 2018).

A fim de entender o funcionamento de organismos no nível molecular, é necessário saber quais são os genes expressados, quando e onde no organismo e seus níveis de expressão. A expressão gênica é um processo complexo regulado em diferentes níveis da síntese proteica (MCCALL, 2013).

Todas as atividades celulares são controladas por seus genes através de uma complexa rede que forma proteínas a partir do DNA (Ácido Desoxirribonucleico - do inglês *Desoxy-ribonucleic Acid*) ao longo de três fases principais: replicação do DNA, transcrição e tradução WATSON *et al.* (2015). A expressão gênica varia ao longo do tempo e depende da relação dos genes nesta rede. A degradação de proteínas e produtos RNA (Ácido Ribonucleico - do inglês *Ribonucleic Acid*) intermediários também podem ser regulados na célula. Tal rede é denominada Rede de Regulação Gênica (GRN - do inglês *Gene Regulatory Network*) (JONG, 2002).

Os modelos de GRNs auxiliam no estudo de fenômenos biológicos mais facilmente e sua inferência a partir de dados de expressão gênica é um problema amplamente abordado em Biologia Sistêmica (ANDRADE, 2012; KLIPP *et al.*, 2016). Esse problema é motivado pela premissa de que um estado funcional de um organismo celular é amplamente determinado pela expressão gênica, baseando-se no Dogma Central da Biologia Molecular (ver Seção 2.1). Enquanto cada uma das células de um organismo complexo contém o mesmo DNA, diferentes genes podem ser transcritos em RNA e traduzidos em proteínas. Isso permite que células de diferentes tecidos do corpo realizem tarefas diferentes e uma célula altere seu comportamento em resposta a estímulos.

Elucidar as relações entre genes e os produtos que eles codificam permanece como um dos desafios centrais da biologia experimental e computacional (JACKSON *et al.*, 2020). Tais relações são utilizadas para descrever e predizer as dependências entre entidades moleculares que resultam em diversos usos práticos, como na identificação de fármacos para o tratamento de doenças, como o câncer (MCCALL, 2013).

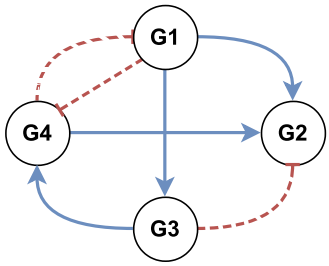
Entretanto, a regulação de um gene não é realizada diretamente. O regulador codifica uma proteína que realiza a regulação. Proteínas específicas, denominadas fatores

de transcrição, ligam-se a uma sequência específica de DNA e aumentam ou diminuem a transcrição de um gene, isto é, o nível ou intensidade da expressão desse gene. Outras proteínas regulam a expressão gênica sem se ligar ao DNA, tais como as envolvidas no remodelamento da cromatina, acetilação ou metilação. Todas essas interações resultam em mudanças na acessibilidade das regiões do DNA e, portanto, mudanças na expressão gênica (MCCALL, 2013).

A maior parte das redes de regulação gênica envolve muitos componentes conectados através de laços de realimentação e relações de autorregulação. Um entendimento intuitivo de sua dinâmica é difícil de obter. Como consequência, métodos formais e ferramentas computacionais para a modelagem e simulação de redes de regulação gênica tornam-se indispensáveis (JONG, 2002), tendo em vista que a compreensão do surgimento de padrões complexos de comportamento a partir das interações entre os genes representa um enorme desafio científico com retornos industriais potencialmente altos (MCCALL, 2013).

Em sua forma mais geral, uma rede consiste em nós e arestas que os conectam. O que os nós e arestas significam determina o tipo de rede. Existem vários tipos de redes celulares, tais como redes metabólicas, redes de sinalização celular e GRNs (ALM; ARKIN, 2003). Nas representações de GRNs, genes são os nós da rede e as relações regulatórias entre os genes são as arestas. A Figura 1 apresenta uma GRN com 4 genes (G1, G2, G3 e G4) onde linhas azuis com setas indicam ativação e vermelhas com barras representam inibição.

Figura 1 – Ilustração de uma GRN composta por quatro genes: G1, G2, G3 e G4. Linhas com setas azuis cheias indicam relação regulatória de ativação entre dois genes enquanto linhas com barras vermelhas tracejadas indicam inibição. Neste exemplo, G1 ativa G2 e G3 e inibe G4. O Gene G3 ativa G4 e inibe G2, e G4 ativa G2 e inibe G1.



Fonte: Elaborado pelo autor (2022).

Existem diversos tipos de modelos de GRNs, dentre contínuos e discretos, sendo simulados de forma determinística ou estocástica. Tipicamente, modelos contínuos são representados por um sistema de equações diferenciais ordinárias enquanto modelos discretos podem ser representados por redes Booleanas, que fornecem uma medida qualitativa dos mecanismos regulatórios (SANGUINETTI *et al.*, 2019). Estes modelos Booleanos, apesar de sua simplicidade, podem representar diversos fenômenos biológicos significativos através de sua dinâmica. Além disso, apesar de modelos contínuos terem sido amplamente

aplicados (MCCALL, 2013), seu uso é limitado para modelar sistemas biológicos onde os parâmetros cinéticos são desconhecidos. A determinação da forma do modelo é uma tarefa árdua e requer etapas posteriores para determinação e validação de tais parâmetros. A escolha de um modelo de rede é comumente baseada no tipo dos dados disponíveis (MCCALL, 2013). Os dados biológicos devem ser analisados para que a estrutura da rede (interações entre os componentes) e os parâmetros do modelo (intensidade e tipo de interação) possam ser aprendidos a partir deles.

Em relação ao tipo de dados, diversas tecnologias de extração/perfilamento podem ser utilizados, tais como *microarrays*, *bulk* RNA-Seq e *single-cell* RNA-Seq (scRNA-Seq). Enquanto a tecnologia *microarray* é capaz de quantificar centenas ou milhares de transcritos gênicos de uma dada célula ou amostra de tecido simultaneamente, o RNA-Seq utiliza tecnologia de sequenciamento de próxima geração para estudar o transcriptoma inteiro. Tecnologias *bulk* RNA-Seq têm sido amplamente utilizadas para estudar os padrões de expressão gênica a nível populacional nas duas últimas décadas (MARIONI *et al.*, 2008; LUO *et al.*, 2021). Contudo, o advento do scRNA-Seq, diferentemente dos métodos de sequenciamento anteriores, proveu oportunidades para explorar os perfis de expressão gênica a nível de uma única célula, possibilitando a interrogação compreensiva e imparcial de cada célula e sua caracterização dos complexos processos biológicos em nível celular. Isso torna a tecnologia de scRNA-Seq favorável para estudar questões centrais de biologia tais como a heterogeneidade celular, descobrimento de novas subpopulações de células importantes na diferenciação celular, entendimento de processos de doenças e o desenvolvimento de embriões iniciais, uma vez que o *bulk* RNA-Seq reflete principalmente a expressão gênica média em milhares de células (CHEN; MAR, 2018; CHEN; NING; SHI, 2019).

Atualmente, um desafio é encontrar GRNs que controlam a diferenciação celular e conduzir transições de um tipo de célula para outro. Contudo, ainda não está claro se GRNs específicas e robustas para estados celulares estáveis podem ser determinados (AIBAR *et al.*, 2017). Isso permanece um desafio principalmente devido à natureza estocástica da expressão gênica. Além disso, os dados de scRNA-Seq possuem características que apresentam dificuldades na análise, interpretação e extração de conhecimento, tais como variação célula-para-célula no sequenciamento profundo, a grande dispersão causada por *dropouts*, efeitos relacionados ao ciclo celular e inexistência de noção física do tempo (KHARCHENKO; SILBERSTEIN; SCADDEN, 2014; BUETTNER *et al.*, 2015). Os experimentos de scRNA-Seq são atrativos para a modelagem de GRNs, uma vez que produzem milhares de medidas independentes (LIU; TRAPNELL, 2016), e os algoritmos de inferência de *pseudotime* ordenam as células ao longo de trajetórias que descrevem o desenvolvimento ou progresso da célula fornecendo uma visão pseudotemporal da cinética da expressão gênica (HAGHVERDI *et al.*, 2016; QIU *et al.*, 2017; SETTY *et al.*, 2016; TRAPNELL *et al.*, 2014).

Juntamente com o aumento da quantidade de dados biológicos, a necessidade para

sua análise levou à proposição de novos algoritmos para a reconstrução de GRNs (DELGADO; GÓMEZ-VELA, 2019). Os algoritmos de inferência de GRNs auxiliam na obtenção de informações sobre fenômenos biológicos e permitem uma exploração mais profunda dos mecanismos regulatórios, podendo, ainda, utilizar os resultados de um algoritmo de inferência para a investigação em laboratório. Desta forma, diversos métodos podem ser encontrados na literatura (IRRTHUM *et al.*, 2010; GAO *et al.*, 2018; MOERMAN *et al.*, 2019; MATSUMOTO *et al.*, 2017; PRATAPA *et al.*, 2020). Entretanto, a maioria dos métodos não se aproveitam explicitamente das características dos tipos de dados envolvidos, tais como as altas taxas de *dropouts* e a inexistência de tempo físico. Além disso, até mesmo os métodos desenvolvidos especificamente para lidar com scRNA-seq não se provaram significativamente superiores aos que usam *microarray* e *bulk* scRNA-Seq (CHEN; MAR, 2018).

Outro fator importante é que não existiam conjuntos de dados amplamente aceitos e critérios de avaliação bem estabelecidos para mensurar a qualidade dos algoritmos quando utilizando de dados scRNA-Seq. Por este motivo, PRATAPA *et al.* (2020) apresentam um conjunto de problemas *benchmark* com variados números de células, genes e taxas de *dropout*, e um *framework* de avaliação das redes inferidas nos quais é possível padronizar as avaliações e comparar diferentes métodos para a inferência de GRNs.

Técnicas de computação evolucionista também tem sido aplicadas para inferir GRNs, tais como Programação Genética e Estratégias Evolutivas (MA *et al.*, 2019; STREICHERT *et al.*, 2004). Contudo, métodos de inferência de GRNs Booleanas e aqueles baseados em técnicas de computação evolucionista não estão listados dentre os algoritmos estado da arte.

Modelos discretos Booleanos são similares aos circuitos digitais, permitindo o uso da Programação Genética Cartesiana (CGP - do inglês *Cartesian Genetic Programming*) (MILLER; THOMSON; FOGARTY, 1997), que é apontada como a técnica evolutiva mais eficiente para o projeto evolutivo de circuitos digitais (VASICEK, 2015, 2018; SOUZA *et al.*, 2020). Desta forma, introduzindo os genes de interesse como entradas, é possível encontrar uma rede Booleana na forma de um circuito digital que represente os ativadores e inibidores desses genes (SILVA *et al.*, 2020). Além disso, redes Booleanas necessitam de dados discretizados, o que leva à uma perda de informação e seu resultado é crítico para o sucesso do processo de inferência.

Por fim, uma vez que a GRN foi inferida, é necessário determinar a qualidade dessa solução. Isso geralmente é feito comparando-se as relações regulatórias inferidas com as relações regulatórias presentes na rede *ground-truth*, quando conhecida, ou utilizando-se de redes de referência obtidas a partir de coleções de experiências biológicas, podendo levar em consideração o tipo celular envolvido. Em dados experimentais, é comum que não se conheça a rede *ground-truth*. Desta forma, problemas sintéticos e suas respectivas redes

ground-truth são comumente utilizadas para testar algoritmos de inferência. Contudo, como ressaltado anteriormente, o desconhecimento da rede *ground-truth* em dados experimentais torna-se um complicador e a literatura relata o baixo desempenho dos algoritmos quando tais dados são considerados.

A avaliação de uma GRN pode ser entendida como um classificador binário. A presença ou ausência das relações regulatórias em relação à rede de referência permite valorar a qualidade da solução levando-se em consideração as métricas associadas à essa classe de problemas, tipicamente contendo dados desbalanceados. Contudo, a literatura não apresenta senso comum no conjunto de métricas a ser utilizado nesta tarefa, bem como não fornece informações sobre a escolha da rede de referência apropriada quando mais de uma está disponível. Além disso, é comum desconsiderar autorregulações no processo de avaliação. Entretanto, a justificativa para isso jaz na natureza dos métodos de inferência, tendo em vista que muitos destes métodos utilizam coeficientes de correlação para determinar possíveis reguladores. Dessa forma, alguns métodos sempre atribuem alta probabilidade de ocorrência de autorregulação enquanto outros a ignoram. Esse fato vai em direção oposta ao relatado pela biologia no que diz respeito à importância de tal sistema de regulação para a manutenção da vida em diversos organismos.

Desta forma, o presente trabalho propõe e apresenta estratégias para a inferência de GRNs a partir de dados de expressão gênica na forma de séries temporais utilizando técnicas de computação evolucionista. Avalia-se os problemas comuns reportados na literatura associados a discretização quando inferindo GRNs Booleanas. As etapas do *framework* proposto são analisadas no que diz respeito aos seus impactos e importância para o processo de inferência de GRNs levando em consideração as características dos tipos de dados envolvidos. Disserta-se sobre possíveis motivos pelos quais algoritmos baseados em computação evolucionista para a inferência de modelos Booleanos não estão contemplados dentre os algoritmos estado da arte. Essas análises culminam em uma revisão da metodologia do processo de inferência e avaliação de GRNs, que discute desde a natureza dos dados utilizados até os critérios de avaliação de GRNs inferidas, propondo sugestões para uma metodologia mais robusta para pesquisadores que lidam com a inferência de GRNs.

Todas as propostas são avaliadas em problemas *benchmark*, considerando tanto dados sintéticos quanto reais. Consideram-se, também, dados oriundos de simulação estocástica e dados de organismos amplamente conhecidos e explorados na literatura, como *Saccharomyces cerevisiae* e *Escherichia coli*, além de dados da competição DREAM4[1]. Além disso, todos os dados reais são resultantes de perfilamento transcricional por meio de *microarrays* ou scRNA-Seq.

Os experimentos abrangem a as etapas independentes da proposta e avaliação do *fra-*

[1] `https://www.synapse.org/#!Synapse:syn3049712/wiki/74630`

mework proposto, denominado CGPGRN, tanto em dados sintéticos e acurados (SILVA *et al.*, 2023) (Seção **4.2.1**) quanto em dados experimentais (SILVA *et al.*, 2024) (Seção **4.2.2**). Foram estudadas alternativas apropriadas de pré-processamento para a proposta (SILVA *et al.*, 2021) (Seção **4.6.1**). Um estudo da discretização apropriada à proposta é realizado (SILVA *et al.*, 2021) (Seção **4.6.3**) e a proposta de um novo método de discretização de dados de expressão gênica é apresentado e discutido (SILVA *et al.*, 2024) (Seção **3.3.1**). Experimentos preliminares também foram realizados a fim de analisar o operador de mutação mais apropriado à proposta no processo de busca da CGP (SILVA *et al.*, 2021) (Seção **4.6.4**). Propõe-se também o uso de paralelismo em GPU para a inferência de modelos da proposta a fim de contornar os problemas de escalabilidade (PRACHEDES *et al.*, 2022a,b). A proposta de obtenção de um modelo contínuo a partir de um modelo Booleano na forma de um sistema de equações diferenciais ordinárias e o ajuste dos coeficientes numéricos via Estratégias Evolutivas (SILVA *et al.*, 2020) é apresentada (Seção 4.5). Além disso, propõe-se um novo processo metodológico para a inferência e avaliação de GRNs (SILVA *et al.*, 2024). Investigações sobre o mecanismo de busca da CGP também foram realizadas (SILVA *et al.*, 2022).

1.1 OBJETIVOS

O objetivo geral deste trabalho é propor e analisar estratégias para a inferência de GRNs a partir de dados de expressão gênica na forma de séries temporais utilizando técnicas de computação evolucionista levando em consideração as especificidades do tipo dos dados.

Para alcançar esse objetivo geral, os seguintes objetivos específicos também são almejados:

- fornecer um *framework* para a inferência e avaliação de GRNs contando com a obtenção de três modelos:
  - um Booleano qualitativo,
  - um contínuo na forma de um sistema de equações diferenciais ordinárias com coeficientes numéricos indefinidos, e
  - um final na forma de um sistema de equações diferenciais ordinárias.
- explorar o uso de metaheurísticas tanto no processo de inferência quanto na determinação de coeficientes numéricos das equações diferenciais ordinárias,
- apresentar um estudo sobre o impacto e a eficiência de diversos métodos de discretização quando aplicados ao problema de inferência de GRNs,
- estudar e avaliar o impacto dos operadores de variação genética na CGP,

- apresentar um novo procedimento de discretização para dados de expressão gênica,
- validar os métodos propostos com experimentos computacionais que abrangem problemas *benchmark* e dados de organismos amplamente estudados, como *S. Cerevisiae* e *E. Coli*,
- propor e apresentar um processo metodológico sistemático para a inferência de GRNs utilizando dados scRNA-Seq,
- disponibilizar os códigos desenvolvidos na forma de um produto (*software*).

1.2 ORGANIZAÇÃO DO TEXTO

O restante deste trabalho está dividido como segue. O Capítulo 2 apresenta os fundamentos necessários de biologia molecular, dogma central da biologia molecular, biologia sistêmica, circuitos lógicos, inferência e solução numérica de equações diferenciais e computação evolucionista, além da definição formal do problema, técnicas de agrupamento, métodos *ensemble*, o uso de computação de alto desempenho em unidades de processamento gráfico, a modelagem e inferência de GRNs e os métodos computacionais utilizados para tal fim, as métricas de avaliação da qualidade das redes inferidas e uma discusão sobre técnicas de agrupamento aplicadas no contexto de biologia sistêmica e uma revisão do processo metodológico de inferência de GRNs utilizando dados perfilados por scRNA-Seq. O Capítulo 3 apresenta o método proposto. O Capítulo 4 apresenta os experimentos computacionais para validação do método proposto. Por fim, o Capítulo 5 apresenta as conclusões e os possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos e fundamentos de biologia molecular relacionados ao problema de inferência de redes de regulação gênica, com foco no dogma central da biologia molecular e regulação gênica. Apresentam-se também, conceitos importantes sobre a área de Biologia Sistêmica inerentes ao processo de inferência de redes de regulação gênica. Conceitos básicos sobre o funcionamento e interpretação de circuitos lógicos, utilizados nos modelos Booleanos, e a inferência e a solução numérica de equações diferenciais, para os modelos contínuos, são brevemente discutidos. São apresentados os fundamentos de computação evolucionista com foco em programação genética cartesiana (CGP) e estratégias evolutivas (ES), que constituem as metaheurísticas utilizadas no procedimento desenvolvido neste trabalho.

Além disso, a caracterização do problema, definição de problema de otimização, especificidades sobre a modelagem e inferência de GRNs, com foco nos métodos de discretização necessários para a inferência de modelos Booleanos, as métricas de avaliação utilizadas para determinação da qualidade das GRNs inferidas e uma discussão sobre técnicas de agrupamento aplicadas no contexto de GRNs são apresentados.

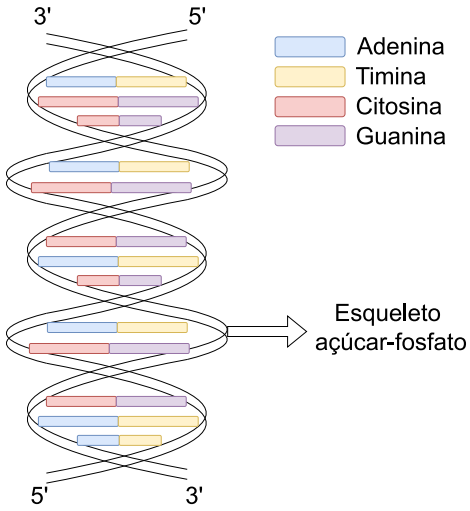
2.1 DOGMA CENTRAL DA BIOLOGIA MOLECULAR

Biologia molecular é o ramo da biologia que almeja entender as bases moleculares das atividades biológicas e das células, incluindo a síntese molecular, modificação, mecanismos e interações (ALBERTS *et al.*, 2010). Todos os seres vivos, animais e vegetais são constituídos por células, que podem ser formalmente definidas como a menor porção de matéria viva dotada e autoduplicação independente (PASSARGE, 2009). As células podem ser classificadas como eucariontes ou procariontes. A principal diferença entre elas é ter, ou não, um núcleo definido. Por conseguinte, o material genético em eucariotos localiza-se no núcleo, enquanto nos procariotos, disperso no citoplasma.

Existem dois tipos de ácidos nucleicos: o ácido desoxirribonucleico (DNA - do inglês *desoxyribonucleic acid*), onde o código genético é armazenado, e o ácido ribonucleico (RNA - do inglês *ribonucleic acid*), que pode desempenhar diversas funções, como participar da síntese proteica, por exemplo. O DNA codifica a informação hereditária de um organismo completo e também é responsável pelo controle do crescimento e divisão celular.

WATSON; CRICK *et al.* (1953), responsáveis pela descoberta da estrutura molecular do DNA, concluíram que o DNA consiste em duas cadeias de ácidos nucleicos, com o esqueleto açúcar-fosfato na parte externa e as bases na parte interna. As cadeias são unidas por ligações hidrogênio entre as bases de uma cadeia e as bases da outra cadeia. As cadeias de DNA são enroladas dentro de uma hélice em torno de um eixo comum. Os pares de bases são planares e paralelos em relação um ao outro na parte interna da

Figura 2 – Representação da dupla-hélice de DNA e as características de antiparalelismo e complementariedade.



Fonte: Adaptado de ALBERTS *et al.* (2010) (2022).

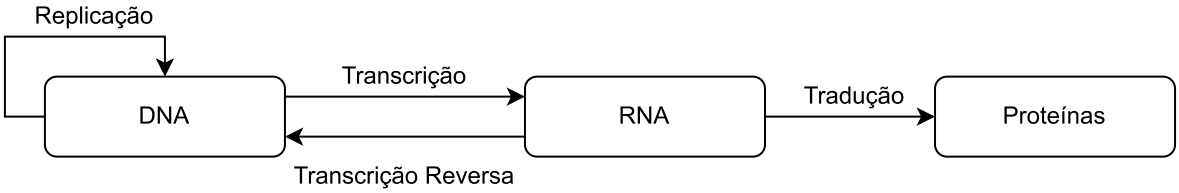
hélice (ALBERTS *et al.*, 2010). Isso constitui as duas principais características do DNA: antiparalelismo e complementariedade (WATSON *et al.*, 2015; ALBERTS *et al.*, 2010). Enquanto a primeira diz respeito à disposição em direções opostas das fitas de DNA, que por convenção é escrita na direção 5' → 3', a segunda reforça que a 5' de uma fita é ligada à 3' da outra, e vice-versa. Além disso, a estrutura primária do DNA é a sequência de bases presentes na cadeia, enquanto a estrutura secundária é a própria dupla-hélice. A representação da estrutura do DNA é apresentada na Figura 2. Por convenção, a sequência de bases em um polinucleotídeo é escrita na direção 5'→3'.

Existem somente quatro bases no DNA, denominadas Adenina (A), Guanina (G), Citosina (C) e Timina (T) (BRUICE, 2006). Já no RNA, a Timina é substituída pela Uracila (U). Os experimentos realizados por Erwin Chargaff (CHARGAFF; LIPSHITZ; GREEN, 1952; ELSON; CHARGAFF, 1952) levaram à conclusão de que a adenina sempre se emparelha com a timina e que a guanina sempre se emparelha com a citosina.

Na maioria dos organismos, a informação genética estocada no DNA é transcrita para o RNA. Essa informação pode, então, ser traduzida para a síntese de todas as proteínas necessárias para a estrutura e função celulares (BRUICE, 2006) se o RNA transcrito for o mRNA (RNA mensageiro). O fluxo que segue do DNA até a proteína é conhecido como Dogma Central da Biologia Molecular, introduzido por Francis Crick (CRICK, 1958).

Este dogma reside no fato de que uma vez que a informação é passada para uma proteína, ela não pode ser trazida de volta. Em mais detalhes, a transferência de informação de ácido nucleico (DNA) para ácido ribonucleico (RNA), e vice-versa, ou de ácido ribonucleico para proteína é possível mas a transferência de proteína para proteína ou de proteína para ácido nucleico/ribonucleico é impossível, conforme ilustrado na Figura 3.

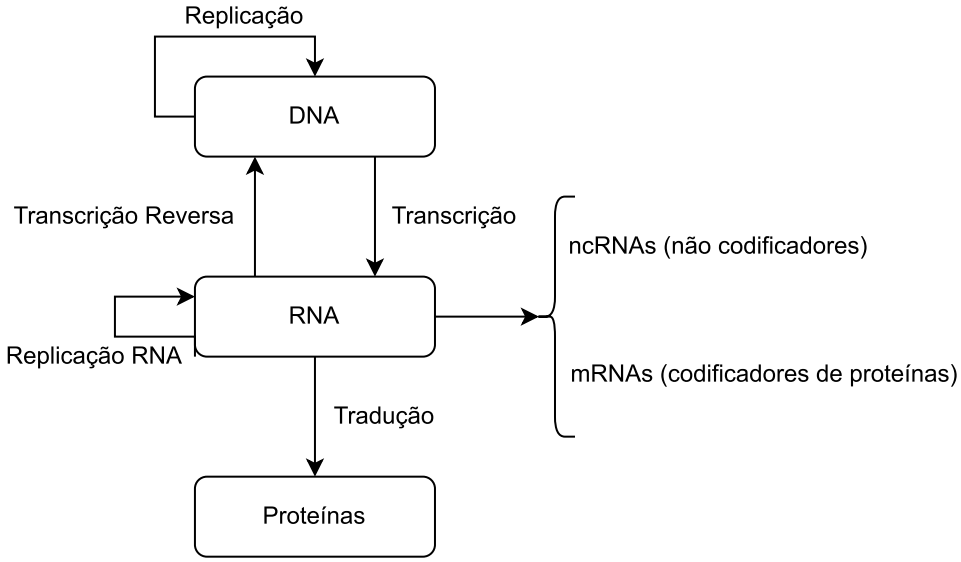
Figura 3 – Dogma Central da Biologia Molecular.



Fonte: Elaborado pelo autor (2022).

Entretanto, este modelo do dogma foi modificado conforme houve progresso na área de biologia molecular. Descobriu-se então que o RNA também pode ser replicado e que os RNAs podem ser classificados em não codificadores e codificadores de proteínas. Este novo modelo, denominado Dogma Central Ampliado, é apresentado na Figura 4.

Figura 4 – Dogma Central Ampliado da Biologia Molecular.



Fonte: Elaborado pelo autor (2022).

Nas próximas subseções são discutidos os fluxos do dogma central da biologia molecular.

2.1.1 Replicação do DNA

A proposta de Watson e Crick para a estrutura do DNA (CRICK, 1958) apresentou evidências de que o DNA é capaz de passar informação genética para gerações posteriores. Esse processo, denominado replicação do DNA, deve ocorrer antes de a célula produzir duas células-filhas geneticamente iguais.

O processo de replicação é feito através do uso de um DNA-molde, num processo pelo qual a sequência de nucleotídeos e uma fita é copiada em sequência complementar de DNA e pode ser resumido nos seguintes passos (ALBERTS *et al.*, 2010; WATSON *et al.*,

2015): (i) Abertura da dupla-hélice, separando a hélice de DNA em duas fitas-molde, (ii) Síntese do DNA das duas fitas separadas através de um complexo multienzimático que contém a DNA-polimerase.

2.1.2 Transcrição

Todo o RNA de uma célula é produzido a partir de uma sequência de DNA, em um processo denominado transcrição, que apresenta certas similaridades em relação ao processo de replicação do DNA (WATSON *et al.*, 2015). A área responsável pelo estudo da transcrição é denominada transcriptômica. O conjunto de todos os transcritos é denominado transcriptoma.

Segmentos individuais da longa sequência de DNA são transcritos em moléculas de mRNA, codificando diferentes proteínas. Cada um desses segmentos de DNA representa um gene (ALBERTS *et al.*, 2010).

A transcrição também é realizada com o auxílio de enzimas. Em seres procariotos, um único tipo de RNA-polimerase é utilizada. Entretanto, em seres eucariotos, o processo envolve diversas enzimas, dentre elas três tipos de RNA-polimerase (I, II e III), e acontece no núcleo da célula (CLARK; PAZDERNIK, 2013).

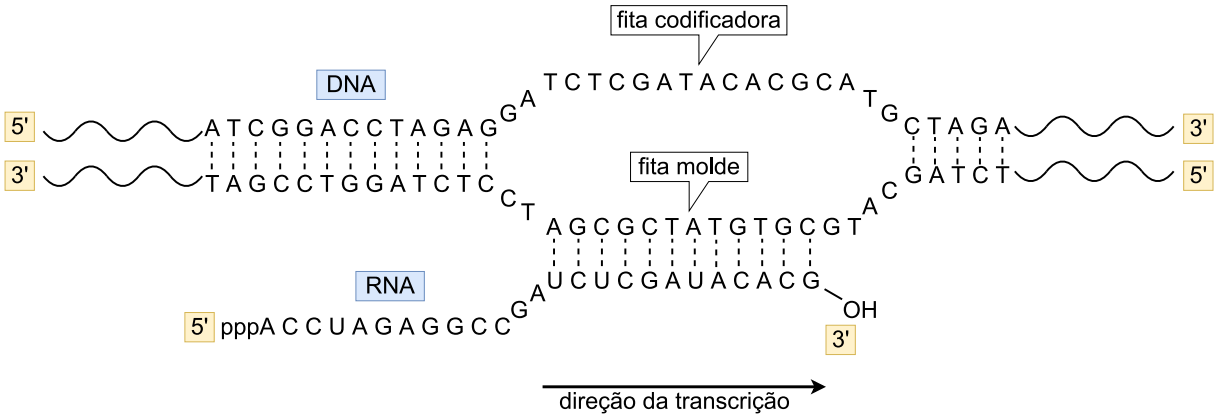
Além disso, proteínas denominadas fatores de transcrição auxiliam na ligação entre a RNA-polimerase e o DNA a fim de iniciar a transcrição (WATSON *et al.*, 2015). Enquanto a RNA-polimerase bacteriana requer apenas um único fator de transcrição (σ), as RNA-polimerases eucarióticas exigem muitos desses fatores, chamados coletivamente de fatores gerais de transcrição (ALBERTS *et al.*, 2010).

É importante destacar que o DNA das células eucarióticas está empacotado em nucleossomos, os quais ainda são organizados em estruturas de cromatina de maior complexidade. Por isso, a iniciação da transcrição nas células eucarióticas requer mais proteínas do que a iniciação da transcrição em células procarióticas. Essas proteínas, conhecidas como ativadoras transcricionais, ligam-se à sequências específicas sobre o DNA (denominadas *enhancers*), auxiliando na atração da RNA-polimerase II para o ponto de iniciação da transcrição.

Conforme apresentado na Figura 5, o processo de transcrição inicia-se quando a RNA-polimerase liga-se ao sítio promotor, uma região que marca o início da transcrição do gene. Nela, a dupla hélice do DNA é desenrolada e formam-se duas cadeias simples, expondo as bases. Uma das cadeias simples é denominada fita codificadora e a outa, fita molde. As bases da fita molde especificam as bases que precisam ser incorporadas ao RNA, seguindo o mesmo emparelhamento das bases encontrado no DNA, exceto pela Timina (T) que é substituída pela Uracila (U).

O RNA produto da transcrição, comumente chamado de pré-mRNA ou transcrito primário, é o precursor de todos os RNAs seguintes.

Figura 5 – Representação do processo de transcrição. A fita molde é a fita de DNA lida na direção 3'→5', consequentemente, o RNA é sintetizado na direção 5'→3'.



Fonte: Adaptado de ALBERTS *et al.* (2010) (2022).

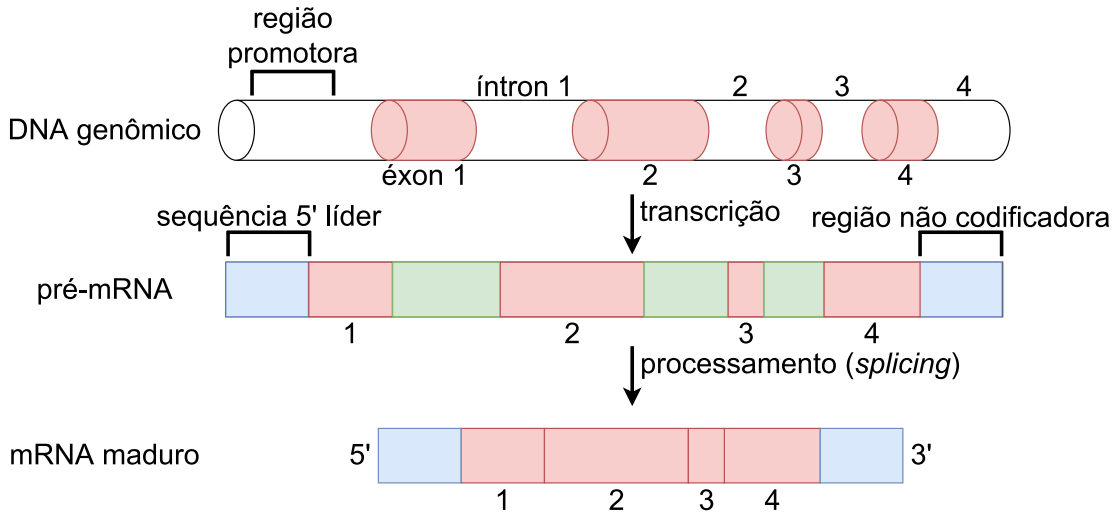
Tais moléculas de RNA, assim como as proteínas, servem como componentes estruturais, reguladores e enzimáticos para uma ampla gama de processos na célula (CLARK; PAZDERNIK, 2013). Algumas moléculas de pequenos RNAs nucleares (snRNA, *small nucelar RNA*) promovem o *splicing* (excisão de íntrons) do pré-mRNA para formar o mRNA. Moléculas de RNA ribossômico (rRNA) formam a porção central dos ribossomos e moléculas de RNA transportador (tRNA) formam os adaptadores que selecionam aminoácidos e os colocam no local adequado nos ribossomos para serem incorporados em polipeptídeos. Moléculas de micro-RNA (miRNA) e moléculas de pequenos RNAs de interferência (siRNA) atuam como importantes reguladores na expressão gênica em eucariotos, e os RNAs que interagem com piRNAs protegem linhagens germinativas animais da ação dos elementos de transposição.

Um gene não é necessariamente uma sequência contínua de bases. Em geral, as bases de um gene são interrompidas por bases que parecem não ter conteúdo informacional. O comprimento de bases que representa uma porção de um gene é chamado éxon, enquanto o comprimento de bases que não expressa informação genética é chamado íntron (WATSON *et al.*, 2015; ALBERTS *et al.*, 2010). Após a síntese do RNA, mas antes de deixar o núcleo no caso de eucariotos, as bases presentes nos íntrons são eliminados e os éxons são unidos, resultando em uma molécula de RNA muito menor, denominada mRNA maduro (WATSON *et al.*, 2015; BRUICE, 2006; WEAVER, 2011). Esse processo, denominado *splicing* é apresentado na Figura 6.

2.1.3 Tradução

Uma vez finalizada a transcrição do mRNA, a informação presente em sua sequência de nucleotídeos é utilizada para sintetizar uma proteína. A sequência de nucleotídeos de um gene, por intermédio do mRNA é traduzida na sequência de aminoácidos de uma

Figura 6 – Representação do *splicing*. O pré-mRNA é o transcrito primário. O mRNA maduro é formado quando os íntrons são removidos do pré-mRNA.



Fonte: Elaborado pelo autor (2022).

proteína, aplicando-se as regras conhecidas como código genético. A esse processo, dá-se o nome de tradução.

Como o RNA é formado de somente quatro nucleotídeos diferentes, existem mais combinações de nucleotídeos do que aminoácidos diferentes. Dessa maneira, alguns aminoácidos são determinados por mais de um triplete de nucleotídeos (códons) (CLARK; PAZDERNIK, 2013; WEAVER, 2011).

A síntese proteica é realizada no ribossomo, através do pareamento do anticódon (um conjunto de três nucleotídeos consecutivos presentes no RNA transportador - tRNA) com o códon complementar presente no mRNA. Conforme os códons penetram no ribossomo, a sequência de nucleotídeos do mRNA é traduzida na sequência de aminoácidos, usando os tRNAs como adaptadores para adicionar cada aminoácido na sequência correta. Cada novo aminoácido é adicionado à cadeia em crescimento em um ciclo de reações, apresentado de maneira macro na Figura 7.

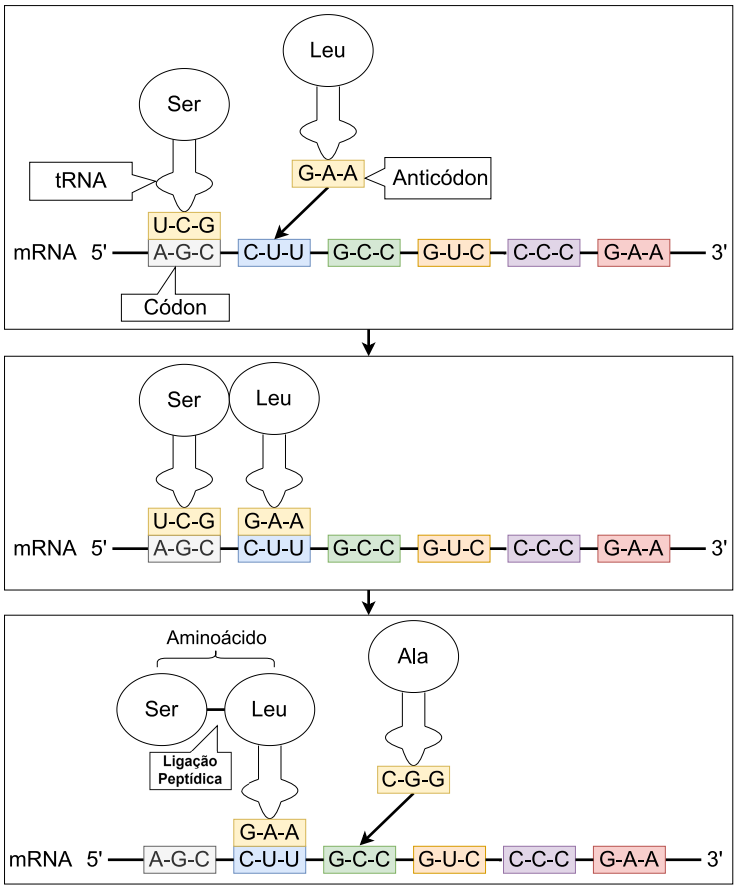
A síntese proteica é finalizada quando um códon de terminação é encontrado. Desta forma, o ribossomo libera sua molécula de mRNA.

2.1.4 Proteínas

As proteínas compõem a maior parte da massa seca de uma célula (WATSON *et al.*, 2015) e são constituídas por cadeias lineares de aminoácidos, unidos por ligações peptídicas, comumente conhecidas por cadeias polipeptídicas (ALBERTS *et al.*, 2010). Elas não são apenas unidades fundamentais das células, mas também executam a maior parte das funções celulares.

Um dos papéis mais importantes das proteínas nas células é catalisar reações

Figura 7 – Visão macro da tradução. Os anticódons ligam-se à códons específicos realizando a síntese do aminoácido que é adicionado à cadeia polipeptídica.



Fonte: Elaborado pelo autor (2022).

bioquímicas, sendo chamadas de enzimas. Além disso, quase todos os processos que ocorrem em uma célula são realizados por proteínas, tais como o controle de moléculas para dentro e para fora da célula, transporte de mensagens de uma célula para outra e integração de sinais. Algumas proteínas especializadas podem atuar como anticorpos, toxinas, hormônios, moléculas anticongelantes, fibras elásticas, fibras de sustentação ou fontes de bioluminescência (WATSON *et al.*, 2015).

Além disso, as proteínas são alvo de diferentes modificações, utilizadas para regular a atividade da proteína, alterando sua conformação, sua ligação a outras proteínas e sua localização na célula. As proteínas desempenham papéis tão importantes que existe uma área específica (proteômica) responsável pelos estudos estruturais e funcionais das proteínas. Entretanto, segundo ALBERTS *et al.* (2010), um passo necessário para o aumento da compreensão sobre as proteínas requer novos métodos bioquímicos, pelos quais pequenos conjuntos de proteínas que interagem entre si possam ser purificados e caracterizados em detalhe. Além disso, novos métodos computacionais serão necessários para a análise do grande volume e complexidade dos dados.

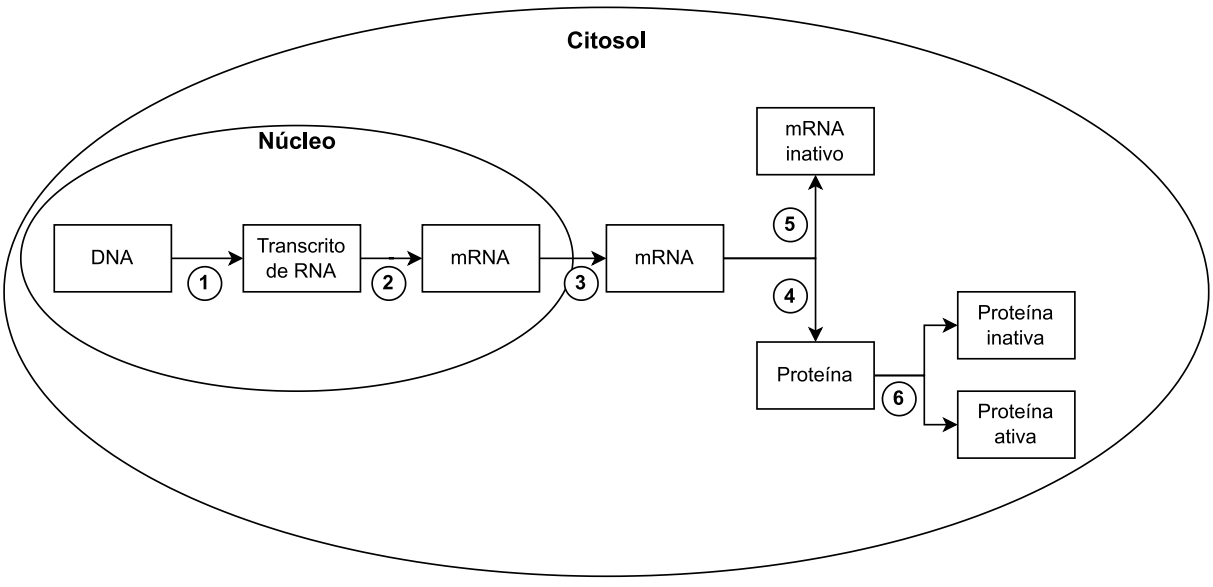
2.2 EXPRESSÃO E REGULAÇÃO GÊNICA

Em procariotos, por conta de sua simplicidade enquanto organismo, o controle da expressão gênica é amplamente estudado e mais bem compreendido (SAURO, 2014). Por outro lado, em eucariotos, os diferentes tipos celulares em um organismo multicelular diferem drasticamente, tanto em estrutura como em função (ALBERTS *et al.*, 2010). Nem todos os genes são expressos em todas as células o tempo todo. Todas as células humanas contêm basicamente os mesmos genes, mas o conjunto de genes expressos para a formação de um tipo celular é diferente do conjunto expresso para a formação de outro (WATSON *et al.*, 2015). A diferença reside no fato das células sintetizarem e acumularem diferentes conjuntos de moléculas de RNA e proteína.

A biologia ainda não sabe dizer quantas diferenças existem entre um tipo celular e outro, mas algumas afirmações gerais podem ser feitas (ALBERTS *et al.*, 2010): (i) muitas células possuem produtos gênicos em comum, (ii) algumas proteínas e RNAs são abundantes nas células especializadas nas quais elas atuam e não podem ser detectadas em nenhum outro local, (iii) a qualquer momento, uma célula humana típica expressa cerca de 30% a 60% dos seus genes em algum nível. Dos aproximadamente 30 mil genes, 21 mil codificam proteínas e 9 mil são RNAs não codificantes, (iv) o nível de expressão de praticamente todos os genes varia de um tipo celular para outro, e (v) existem muitos passos após a produção do RNA nos quais a expressão gênica pode ser regulada.

Em relação ao último item, uma célula eucariótica pode controlar as proteínas que produz de diversas maneiras, conforme apresentado na Figura 8. Os círculos numerados correspondem aos momentos em que o controle pode ocorrer e são descritos a seguir:

Figura 8 – Pontos de controle da expressão gênica no fluxo de DNA até uma proteína.



Fonte: Elaborado pelo autor (2022).

1. controlando quando e como um gene é transcrito (controle transcricional);
2. controlando como o transcrito de RNA é submetido a *splicing* ou é processado (controle do processamento de RNA);
3. selecionando quais mRNAs completos são exportados do núcleo para o citoplasma e determinando sua localização no citoplasma (controle e transporte e da localização de RNA);
4. selecionando quais mRNAs no citoplasma são traduzidos pelos ribossomos (controle traducional);
5. desestabilizando seletivamente certas moléculas de mRNA no citoplasma (controle da degradação do mRNA); ou
6. ativando, inativando, degradando ou compartimentalizando seletivamente moléculas de proteína específicas após sua produção (controle da degradação do mRNA).

Além destes pontos, sinais externos podem induzir uma célula a alterar a expressão de seus genes, chamado de regulação epigenética.

Os genes em sua maioria são regulados em múltiplos níveis. Os mecanismos reguladores incluem (ALBERTS *et al.*, 2010; WATSON *et al.*, 2015) a atenuação do transcrito de RNA pela sua terminação prematura, a seleção de sítios de *splicing* alternativos do RNA, o controle da formação das extremidades 3' por clivagem e adição de poli-A, edição do RNA, o controle do transporte do núcleo para o citosol, a localização dos mRNAs em sítios determinados da célula, o controle do início da tradução e a degradação regulada do mRNA.

A maioria desses processos de controle necessita do reconhecimento de sequências específicas ou de estruturas na molécula de RNA que está sendo regulada que é uma tarefa desempenhada tanto por proteínas reguladoras como por moléculas de RNA reguladoras (ALBERTS *et al.*, 2010).

Para a maioria dos genes, os controles transcricionais são os mais importantes pois somente o controle transcricional garante que a célula não sintetizará intermediários supérfluos além de ser a etapa mais eficiente energeticamente para regular (ALBERTS *et al.*, 2010; WATSON *et al.*, 2015). Em contrapartida, regular etapas posteriores pode apresentar vantagens, principalmente no que diz respeito a reduzir o tempo de resposta da regulação. Nas próximas subseções discute-se brevemente como o controle acontece nas principais etapas, com foco no controle transcricional e a importância dos RNAs reguladores.

2.2.1 Controle Transcricional

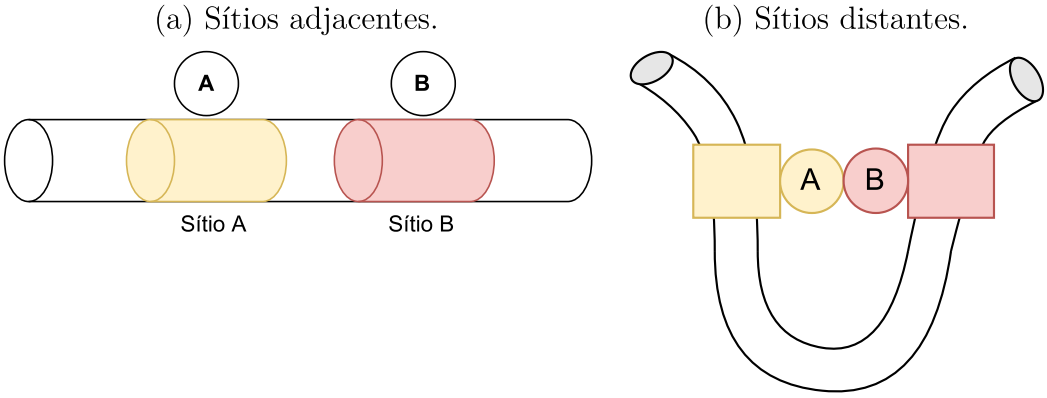
De maneira simplificada, a regulação da expressão gênica transcricional é realizada através da comunicação de proteínas reguladoras aos genes. Os reguladores podem ser positivos (ativadores), aumentando a transcrição do gene alvo, ou negativos (repressores/inibidores), reduzindo a transcrição. Esses reguladores são proteínas de ligação ao DNA que reconhecem sítios específicos nos genes que elas controlam (ALBERTS *et al.*, 2010; WATSON *et al.*, 2015).

O controle transcricional é realizado na etapa de transcrição do DNA em RNA e o procedimento geral é o mesmo: a RNA-polimerase liga-se ao promotor, formando um complexo fechado no qual as fitas de DNA permanecem unidas. Após a ligação, o complexo polimerase-promotor sofre transição para o complexo aberto, em que o DNA do sítio de início de transcrição é desenrolado e a polimerase é posicionada para iniciar a transcrição. No passo seguinte, a polimerase deixa o promotor, inicia a fase de alongamento e a transcrição é iniciada.

Os repressores podem inibir a transcrição ligando-se a um sítio que se sobrepõe ao promotor, bloqueando a ligação da RNA-polimerase. Os repressores também podem atuar de outros modos, por exemplo, pela ligação a um sítio ao lado do promotor e interação com a ligação entre polimerase e promotor, inibindo o início (ALBERTS *et al.*, 2010).

Frequentemente, as proteínas de ligação ao DNA que interagem entre si ligam-se a sítios adjacentes. Entretanto, algumas proteínas interagem mesmo quando ligadas a sítios distantes do DNA, através do dobramento entre os sítios do DNA, como apresentado na Figura 9. A essas interações, dá-se o nome de ligação cooperativa (ALBERTS *et al.*, 2010).

Figura 9 – Ligações cooperativas.



Fonte: Elaborado pelo autor (2022).

Os sítios de ligação de reguladores podem estar agrupados em unidades chamadas reforçadores (*enhancers*). Reforçadores alternativos ligam-se a diferentes grupos de reguladores e controlam a expressão do mesmo gene em diferentes momentos e locais, em resposta

a diferentes sinais. Outras sequências reguladoras, chamadas isoladores, estão localizadas entre os reforçadores e alguns promotores. Sua utilidade é bloquear a ativação do promotor por ativadores ligados ao reforçador (ALBERTS *et al.*, 2010; CLARK; PAZDERNIK, 2013; WEAVER, 2011).

A determinação de quais genes devem ser transcritos é baseado em um grupo de proteínas, responsáveis por reconhecer as sequências reguladoras *cis*-atuantes[1], que são sítios de ligação para os reguladores transcricionais, cuja presença do DNA afeta a taxa de iniciação da transcrição. É importante destacar que a transcrição de cada gene é controlada por seu conjunto particular de sequências reguladoras (WATSON *et al.*, 2015).

Aproximadamente 10% dos genes codificadores de proteínas da maioria dos organismos produzem reguladores transcricionais tornando essa uma das maiores classes e proteínas nas células (ALBERTS *et al.*, 2010). Na maioria dos casos, um dado regulador transcricional reconhece as suas próprias sequências reguladoras *cis*-atuantes, que são diferentes daquelas reconhecidas por todos os outros reguladores presentes na célula.

Ainda que alguns poucos genes sejam controlados por uma única sequência reguladora *cis*-atuante, que é reconhecida por um único regulador transcricional (óperon), a maioria dos genes possui arranjos complexos de sequências reguladoras *cis*-atuantes, cada uma delas sendo reconhecida por um regulador transcricional diferente. Desse modo, as posições, identidade e arranjo das sequências reguladoras *cis*-atuantes determinam, em última análise, o momento e o local em que cada gene é transcrito. Além disso, um regulador pode, com frequência, participar de mais de um tipo de complexo regulador. Desta forma, reguladores transcricionais eucarióticos funcionam como partes reguladoras que são usadas para construir complexos cuja função depende da montagem final de todos os componentes individuais, tornando possível que o mesmo regulador transcricional possa atuar como ativador ou inibidor a depender dos demais componentes do complexo.

Além disso, várias proteínas ativadoras ligadas ao DNA atuando em conjunto produzem uma taxa de transcrição muito superior à soma das taxas de transcrição alcançadas quando atuam individualmente (ALBERTS *et al.*, 2010). A isso se dá o nome de sinergia transcricional.

É possível visualizar quando uma dada proteína se liga a uma região definida do DNA em uma célula através da técnica chamada imunoprecipitação da cromatina (ChIP, *chromatin immunoprecipitation*). Elaborações deste método, como a ChIP-chip e a ChIP-Seq fornecem técnicas para a extração de dados transcriptômicos, brevemente discutido na Seção **2.2.5**.

[1] Sequências que estão no mesmo cromossomo em que se encontram os genes que elas controlam.

2.2.2 Epigenética e Vias de Transdução de Sinal

A regulação epigenética é um tipo de regulação que ocorre quando sinais externos induzem alterações na expressão gênica de uma célula. Esse tipo de regulação é especialmente importante quando os padrões de expressão gênica têm de ser herdados (ALBERTS *et al.*, 2010). Durante o desenvolvimento, um sinal liberado por uma célula provoca a ativação de genes específicos nas células vizinhas. Dessa forma, alguns desses genes permanecem ativos nas células por muitas gerações, mesmo que o sinal que os induziu tenha sido apenas fugaz. Esse tipo de regulação é comumente visualizado na divisão celular, mesmo que o sinal iniciador não esteja mais presente. Frequentemente, os sinais são comunicados para os reguladores transcricionais através de vias de transdução de sinal (WATSON *et al.*, 2015). Os sinais podem ter várias formas, como pequenas moléculas de açúcares, mas podem também ser proteínas liberadas por uma célula e recebidas por outra. Isso é especialmente comum durante o desenvolvimento de organismos pluricelulares. Há vários modos pelos quais os sinais são detectados por uma célula e comunicados para um gene. Em bactérias, os sinais controlam as atividades dos reguladores por meio da indução de alterações alostéricas[2] nestes reguladores. Frequentemente, este efeito é direto: um pequeno sinal molecular, como um açúcar, entra na célula e liga-se diretamente ao regulador transcricional. Entretanto o efeito do sinal pode ser indireto e é chamado de uma via de transdução de sinal. Em uma via de transdução de sinal, o ligante inicial é normalmente detectado por um receptor de superfície celular específico. Desta forma, o ligante liga-se a um domínio extracelular do receptor e esta ligação é comunicada ao domínio intracelular. A seguir, o sinal é retransmitido para o regulador de transcrição relevante.

2.2.3 RNAs Reguladores

Os RNAs reguladores eucarióticos existem em múltiplas formas, caracterizados por seu tamanho ("longos" ou "curtos"), sua origem e os mecanismos pelos quais eles são gerados e regulam a expressão gênica. Acredita-se que entre 30% e 70% dos genes em eucariotos complexos sejam regulados até certo ponto por RNAs, com papéis que vão desde o desenvolvimento até a homeostase celular e proteção das células contra vírus e transposons (WATSON *et al.*, 2015; CLARK; PAZDERNIK, 2013).

Os RNAs gerados a partir de precursores de dsRNA (RNAs de dupla-fita), são chamados de pequenos RNAs de interferência (siRNAs, *small interfering RNAs*)[3]. Outro grupo de RNAs reguladores é o de microRNAs (miRNAs). Estes miRNAs são derivados de RNAs precursores que são codificados por genes expressos em células nas quais estes

[2] Qualquer alteração na estrutura terciária ou quaternária de uma enzima proteica induzida pela ação de uma molécula ligante.

[3] Este tipo de regulação foi adaptada para uso como uma ferramenta experimental para manipular a expressão gênica em vários organismos.

miRNAs possuem funções reguladoras específicas. Uma terceira classe de RNAs reguladores curtos são os RNAs de interação com piwi (piRNAs, piwi-interaction RNAs), que são expressos predominantemente na linhagem germinativa e possuem características distintas das de miRNAs.

Vários tipos de RNAs muito curtos reprimem, ou silenciam, a expressão de genes com homologia a estas sequências curtas de RNA. Dependendo da origem e do contexto, estes RNAs atuam inibindo a tradução do mRNA, destruindo o mRNA, ou mesmo silenciando a transcrição a partir do promotor que dirige a expressão do mRNA (WATSON *et al.*, 2015; ALBERTS *et al.*, 2010). Estes RNAs curtos são geralmente produzidos por enzimas especiais a partir de dsRNAs mais longos.

O exemplo mais bem compreendido da regulação por RNAs longos (com 200 nucleotídeos ou mais) é o *Xist*, que dirige a inativação de um cromossomo X na compensação de dose de mamíferos (ALBERTS *et al.*, 2010).

Os miRNAs são codificados por genes em organismos nos quais eles normalmente atuam como reguladores de genes envolvidos no desenvolvimento (ALBERTS *et al.*, 2010). No caso do genoma humano, mais de mil miRNAs diferentes são produzidos e parecem regular pelo menos um terço de todos os genes codificadores de proteínas (WATSON *et al.*, 2015). Estes pequenos RNAs inibem a expressão de genes-alvo homólogos desencadeando a destruição do mRNA codificado pelo gene-alvo, inibindo a tradução do mRNA ou induzindo modificações da cromatina no gene-alvo, silenciando a transcrição (ALBERTS *et al.*, 2010).

Segundo ALBERTS *et al.* (2010), diversas características tornam os miRNAs reguladores especialmente úteis na expressão gênica. Um único mRNA pode regular um conjunto inteiro de mRNAs diferentes se os mRNAs carregarem uma sequência curta comum. Segundo, a regulação por miRNA pode ser combinatória. Quando o pareamento entre o miRNA e o mRNA falha em desencadear a clivagem, miRNAs adicionais ligando-se ao mesmo mRNA conduzem a reduções maiores na sua tradução. Terceiro, um miRNA ocupa um espaço relativamente pequeno no genoma quando comparado a uma proteína.

Um uso bem compreendido dos RNAs não codificadores ocorre na interferência de RNA (iRNA), na qual os RNAs-guia (miRNA, siRNAs, piRNAs) se pareiam com mRNAs. A iRNA pode induzir os mRNAs a serem destruídos ou terem a sua tradução reprimida. Ela também pode induzir que genes específicos sejam empacotados em heterocromatina suprimindo sua transcrição.

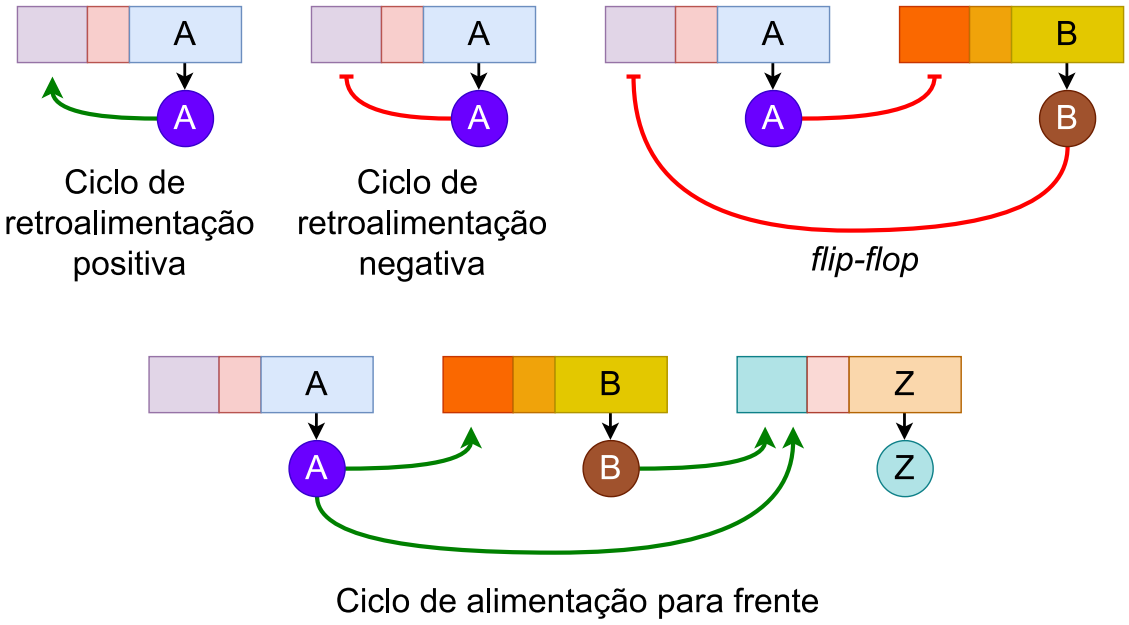
2.2.4 Circuitos de Transcrição

Os mecanismos de regulação gênica podem ser combinados para criar "circuitos lógicos" pelos quais as células integram sinais e relembram eventos passados (ALBERTS *et al.*, 2010). Os circuitos de regulação gênica simples podem ser combinados para criar

todos os tipos de mecanismo de controle.

Segundo ALBERTS *et al.* (2010), uma análise de circuitos de regulação gênica revela que certos tipos simples de arranjo (denominados motivos de rede) são encontrados repetidamente em células de espécies amplamente diferentes. Os tipos mais comuns de *motifs* de rede são apresentados na Figura 10.

Figura 10 – *Motifs* de Rede. A e B representam reguladores transcricionais, setas indicam controle positivo e barras, controle negativo. No ciclo de alimentação para frente, A e B representam reguladores transcricionais que ativam a transcrição do gene-alvo Z.



Fonte: Elaborado pelo autor (2022).

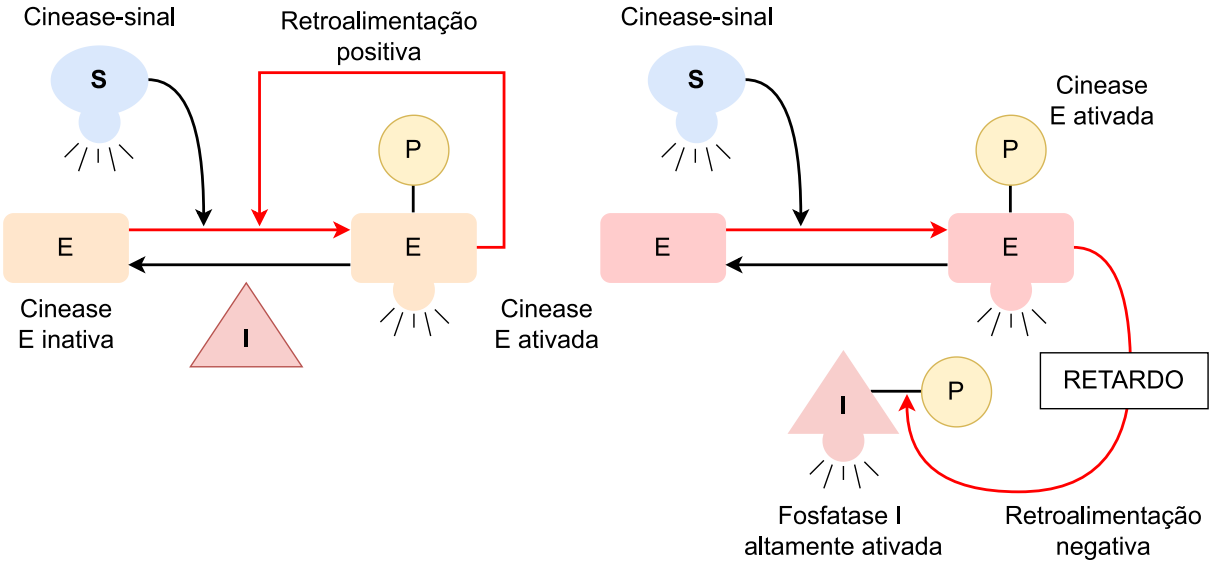
Dentre esses arranjos, ciclos de reatroalimentação, tanto positiva quanto negativa, são comuns em todas as células, como apresentado na Figura 11. Enquanto o ciclo de retroalimentação positiva fornece um mecanismo de memória simples, o negativo com frequência é usado para manter a expressão do gene próximo ao nível padrão, apesar das variações nas condições bioquímicas dentro da célula. Esse mecanismo de memória é um pré-requisito para a criação de tecidos organizados e para a manutenção de tipos celulares estavelmente diferenciados (WATSON *et al.*, 2015).

Os tipos diferentes de comportamento produzidos por um ciclo de retroalimentação irão depender dos detalhes do sistema. Com dois ou mais reguladores transcricionais, a amplitude possível dos comportamentos do circuito torna-se mais complexa.

Cada célula em um organismo multicelular em desenvolvimento é equipada com uma maquinaria e controle similarmente complexa, e deve, de fato, usar seu sistema intrincado de comutadores de transcrição entrelaçados para calcular como ela deve se comportar a cada momento em resposta a muitos estímulos recebidos no passado e no presente. Esses circuitos gênicos podem ser obtidos através de métodos de inferência de

Figura 11 – Retroalimentação.

(a) Retroalimentação positiva. (b) Retroalimentação negativa.



Fonte: Adaptado de ALBERTS *et al.* (2010) (2022).

redes de regulação gênica e serão discutidos em seções posteriores.

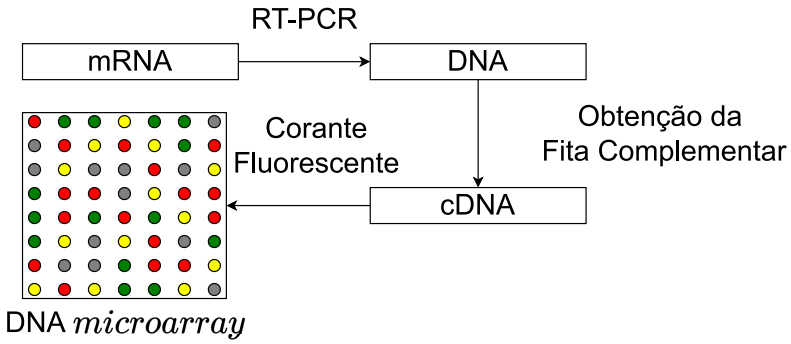
2.2.5 Perfilamento Transcricional

O perfilamento transcricional ou perfilamento da expressão gênica consiste na identificação e caracterização de genes individuais e redes gênicas para melhor compreender a função do gene nos níveis celulares, de tecido e de órgãos em diferentes estados de saúde e doença. O perfilamento da expressão gênica é uma subárea da genômica funcional e os esforços são direcionados para o entendimento das conexões entre a expressão de genes individuais ou grupos de genes e suas funções biológicas únicas. Isso se justifica pelo fato de que apesar de todas as células de um organismo possuírem o mesmo material genético, cada uma é distinguida pelo nível e espectro de ativação ou expressão de um conjunto específico de genes (ALBERTS *et al.*, 2010; WATSON *et al.*, 2015; CLARK; PAZDERNIK, 2013). Neste trabalho, discutiremos três métodos utilizados pelos biólogos experimentais para a quantificação da expressão gênica: *microarrays*, *RNA-Seq*, e *scRNA-Seq*.

Os *microarrays* de DNA podem medir simultaneamente o nível de expressão de milhares de genes dentro de uma amostra particular de mRNA. O processo físico-químico chave envolvido nos *microarrays* é a hibridização do DNA (KNUDSEN, 2005). Duas fitas de DNA hibridizam se elas são complementares uma a outra, de acordo com as leis de Watson-Crick (adenina liga-se à timina e citosina liga-se à guanina).

O mRNA é extraído a partir de tecidos ou células e transcrito reversamente para gerar uma fita de DNA. A segunda fita do DNA (cDNA) é sintetizado e hibridizado, e as regiões de interesse são rotuladas com um corante (usualmente florescente). O passo seguinte consiste em gerar uma imagem utilizando imagem fluorescente induzida por laser

Figura 12 – Processo de obtenção dos *microarrays*. Neste exemplo, consideram-se dois tipos celulares: saudável e doente, marcados com corantes vermelho e verde, respectivamente. No *microarray*, *spots* em vermelho indicam que aquele gene está expressado na célula saudável. Verde indica expressão do gene na célula doente. Amarelo indica expressão do gene em ambas as células e em cinza, não há expressão do gene em nenhuma das células.



Fonte: Elaborado pelo autor (2022).

em cada localização específica das sequências. Esses experimentos não provêm dados acerca do nível absoluto de expressão de um gene particular (concentrações verdadeiras de mRNA), mas são úteis para comparar o nível de expressão entre diferentes condições (células saudáveis e doentes, por exemplo) e entre genes (TARCA; ROMERO; DRAGHICI, 2006). Esse processo é apresentado na Figura 12.

Ainda, segundo (KNUDSEN, 2005), os *microarrays* podem ser amplamente classificados de acordo com pelo menos três critérios (i) comprimento das sondas, (ii) método de fabricação, e (iii) número de amostras que podem ser perfilados simultaneamente em uma matriz. De acordo com o comprimento das sondas, as matrizes podem ser classificadas em "arranjos de cDNA", que usam sondas longas de centenas ou milhares de pares de bases (bps) e "arranjos de oligonucleotídeos", que usam sondas pequenas (geralmente 50bps ou menos). Já os métodos de fabricação incluem a decomposição das sequências previamente sintetizadas e a síntese *in-situ*. Normalmente, as matrizes de cDNA são fabricadas usando decomposição, enquanto as matrizes de oligonucleotídeos são fabricados usando tecnologias *in-situ* (KNUDSEN, 2005). Por fim, o terceiro critério refere-se ao número de amostras que pode ser perfilado em uma matriz. Os *microarrays* de canal único analisam uma única amostra de cada vez, enquanto os *microarrays* de múltiplos canais podem analisar duas ou mais amostras simultaneamente.

Um grande avanço obtido no final dos anos 2000 foi o perfilamento através de RNA-Seq. A eficiência e custo foram melhorados em relação aos *microarrays*. Parte de seu sucesso se deve ao fato de que o RNA-Seq permite uma amostragem imparcial de todos os transcritos em uma amostra, ao invés de se limitar a um conjunto pré-determinado de transcritos, como ocorre em *microarrays* ou RT-qPCR (PREDEU, 2022). Para maiores informações sobre RT-qPCR, refere-se à (CLARK; PAZDERNIK, 2013).

Normalmente, o RNA-Seq é usado em amostras compostas por uma mistura de

células, também conhecido como *bulk* RNA-Seq. Suas aplicações incluem a caracterização de assinaturas de expressão entre tecidos em amostras saudáveis/doentes, de tipo selvagem/mutante ou controle/tratadas. Além disso, é utilizado em estudos evolutivos, usando transcriptômica comparativa de amostras de tecidos em diferentes espécies (PREDEU, 2022). Não obstante, é possível utilizar essa tecnologia para encontrar e anotar novos genes, isoformas de genes e outros transcritos.

Entretanto, com *bulk* RNA-Seq só é possível estimar o nível médio de expressão para cada gene em uma população de células, sem levar em consideração a heterogeneidade na expressão gênica entre células individuais dessa amostra. Logo, é insuficiente para estudar sistemas heterogêneos, tais como estudos de desenvolvimento inicial ou tecidos complexos.

Para superar essa limitação, novos protocolos foram desenvolvidos e permitem a aplicação de RNA-Seq em nível de célula única (scRNA-Seq - *single-cell RNA-Sequencing*), com sua primeira publicação em 2009 (SHIROTA; KINOSHITA, 2016). Entretanto, o scRNA-Seq tornou-se mais popular por volta de 2014 (PREDEU, 2022), quando novos protocolos e custos de sequenciamento mais baixos tornaram-na mais acessível.

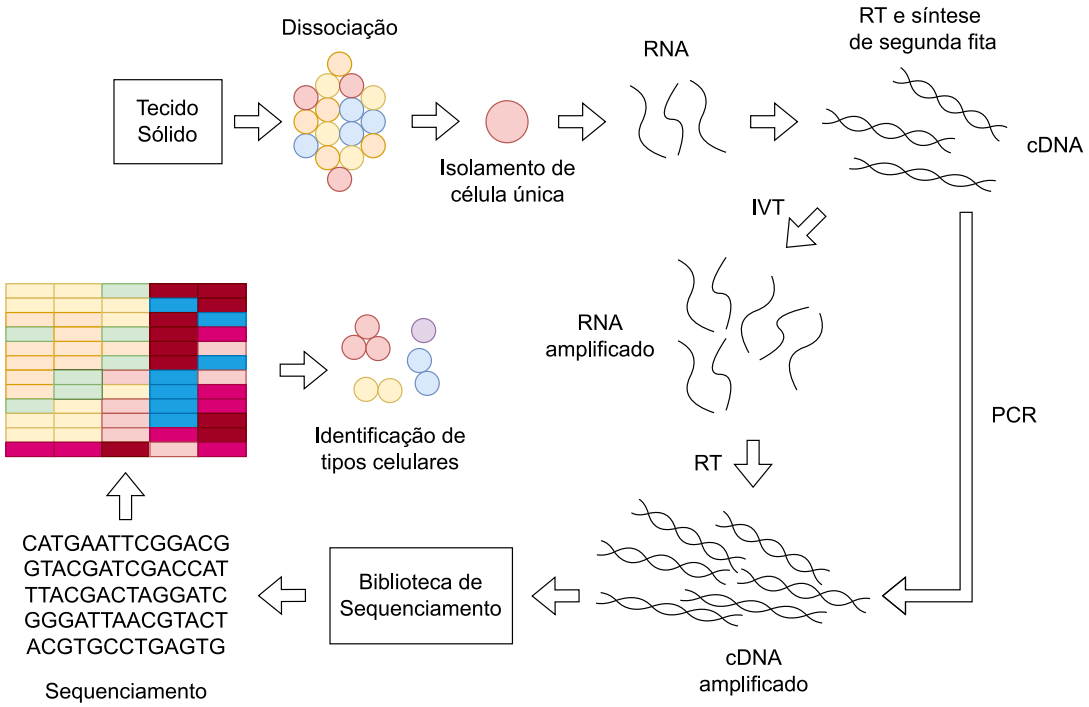
Diferentemente da abordagem em massa, com scRNA-Seq pode-se estimar uma distribuição de níveis de expressão para cada gene em uma população de células. Isso permitiu que novas questões biológicas fossem respondidas, onde as alterações específicas da célula no transcriptoma são importantes. Por exemplo, identificar novos ou raros tipos de células, identificar a composição celular diferencial entre tecidos saudáveis/doentes ou compreender a diferenciação celular durante o desenvolvimento. Segundo Predeu (PREDEU, 2022), um dos usos mais emblemáticos dessa tecnologia é a construção de atlas de genes, que fornecem um compêndio abrangente da diversidade celular em organismos.

Os conjuntos de dados gerados a partir do scRNA-Seq variam de centenas a milhões de células por estudo. Esse número é ainda maior com as tecnologias mais recentes, cada uma com suas vantagens e desvantagens. Desde sua primeira aparição em (SHIROTA; KINOSHITA, 2016), onde apenas uma célula era separada manualmente, protocolos como 10x Genomics e SPLIT-Seq chegam na casa das 100.000 células.

De maneira geral, como apresentado na Figura 13, um protocolo típico de scRNA-Seq segue os seguintes passos (PREDEU, 2022):

1. Dissecção de tecidos e dissociação de células para obter uma suspensão de células;
2. Opcionalmente, as células podem ser selecionadas (por exemplo, com base em marcadores de membrana);
3. Extração do RNA de cada célula;
4. Transcrição reversa do RNA para cDNA mais estável;

Figura 13 – Protocolo scRNA-Seq. (RT - Transcrição Reversa; cDNA - DNA complementar; IVT - transcrição *in vitro*; PCR - Reação em Cadeia Polimerase)



Fonte: Elaborado pelo autor (2022).

5. Ampliação do cDNA (por transcrição *in vitro* ou por PCR);
6. Preparação da biblioteca de sequenciamento com adaptadores moleculares adequados;
7. Sequenciamento, geralmente com protocolos *Illumina* emparelhados;
8. Processamento dos dados brutos para obter uma matriz de contagem de genes por células; e
9. Análises a jusante.

A estratégia utilizada para capturar células determina o rendimento (quantidade de células isoladas) do experimento, como as células são selecionadas antes do sequenciamento, bem como que tipo de informação adicional além do sequenciamento de transcrição pode ser obtida. Segundo PREDEU (2022), as três opções mais utilizadas são: (i) métodos baseados em placas de microtitulação, (ii) baseados em matrizes microfluídicas, e (iii) baseados em gotículas microfluídicas. Os métodos baseados em gotículas microfluídicas são o mais popular atualmente devido ao seu maior rendimento e menor custo (ZIEGENHAIN *et al.*, 2017).

Existem dois tipos de quantificação de transcrição: (i) completo e (ii) baseado em *tags*. Os protocolos completos tentam obter uma cobertura de leitura uniforme em toda a transcrição, enquanto os protocolos baseados em *tags* capturam apenas as extremidades 5'

ou 3'. A escolha do método de quantificação tem implicações para quais tipos de análises os dados podem ser usados.

A preparação de bibliotecas completas para *single-cell* é essencialmente idêntica ao que é feito no *bulk* RNA-seq, e é restrito a protocolos baseados em placas, como SMART-seq2. Embora, em teoria, os protocolos completos devam fornecer uma cobertura uniforme das transcrições, vieses na cobertura em todo o corpo do gene podem ocorrer. Além disso, os protocolos completos também permitem a detecção de variantes de emenda, o que é muito difícil de fazer com outros protocolos (PREDEU, 2022).

Já com os protocolos baseados em *tags*, apenas uma das extremidades (3' ou 5') da transcrição é sequenciada. A principal vantagem desse tipo de protocolo é que eles podem ser combinados com identificadores moleculares exclusivos (UMIs - *unique molecular identifier*), que podem ajudar a melhorar a precisão da quantificação da transcrição. A maioria dos protocolos atuais de *scRNA-Seq* são baseados em *tags* (PREDEU, 2022). Entretanto, uma desvantagem dos protocolos baseados em *tags* é que, sendo restrito a apenas uma extremidade da transcrição, reduz-se a capacidade de alinhar inequivocadamente as leituras a uma transcrição, além de dificultar a distinção de diferentes isoformas (VICKERS, 2017).

É importante destacar que a diferença entre os protocolos baseados em *tags* 5' e 3' é qual extremidade da transcrição é sequenciada. Embora os protocolos 3' sejam mais comumente usados, muitos protocolos permitem o sequenciamento de ambas as extremidades (10x Chromium, por exemplo). A vantagem do sequenciamento 5'-fim é que é possível obter informações sobre o local de início da transcrição (TSS - *transcription start sites*), permitindo explorar se há uso diferencial de TSS entre as células (CLARK; PAZDERNIK, 2013).

Como mencionado anteriormente, a principal diferença entre o *bulk* RNA-Seq e o scRNA-Seq é que cada biblioteca de sequenciamento representa uma única célula, em vez de uma população de células. Isso faz com que não seja possível haverem "replicações" biológicas em nível de célula única, isto é, cada célula é única e impossível de replicar. Contudo, as células podem ser agrupadas por suas similaridades e as comparações podem ser feitas entre grupos de células semelhantes.

Outro grande desafio em scRNA-Seq é que a quantidade de material de partida por célula é muito baixa (PREDEU, 2022). Por esse motivo, os dados são muito escassos levando à uma maioria de genes não detectados e, portanto, os dados contém muitos zeros. Isso gera o problema de identificar se o zero é devido à um gene de fato não estar expressado ou se não foi possível detectá-lo (*dropout*). Consequentemente, é preciso levar em consideração que a variação de uma célula para outra célula nem sempre tem natureza biológica, mas sim problemas técnicos causados pela amplificação de PCR desigual entre células e *dropouts*, que pode ser formalmente definido como a detecção de um gene em uma célula mas não detectado em outra (KHARCHENKO; SILBERSTEIN; SCADDEN, 2014).

Outro aspecto importante a ser considerado são os efeitos de lote (*batch effects*), que podem ser observados mesmo ao sequenciar o mesmo material usando tecnologias diferentes. Se não forem normalizados, vieses são introduzidos. Como o perfilamento é realizado célula à célula, não existe noção física de tempo como nos *microarrays* e *bulk* RNA-Seq. Para que seja possível obter uma noção temporal da expressão gênica das células é preciso estimar o *pseudotime*. O *pseudotime* é uma noção de desenvolvimento biológico da célula e é obtido por técnicas de inferência, geralmente baseadas em *clustering*, como o Slingshot (STREET *et al.*, 2018; PRATAPA *et al.*, 2020). Além disso, na diferenciação celular, por exemplo, uma mesma célula pode possuir *steady-states* diferentes. Esses diferentes *steady-states* podem ser facilmente visualizados com as trajetórias de *pseudotimes*.

2.3 BIOLOGIA SISTÊMICA

Biologia Sistêmica almeja entender sistemas biológicos no nível de sistema e consiste na análise quantitativa da maneira pela qual todos os componentes de um sistema biológico interagem funcionalmente ao longo do tempo (ADEREM, 2005). É uma área crescente na biologia, devido ao progresso de diversos campos, principalmente os no campo da biologia molecular, tais como as tecnologias para reconstrução do DNA, medidas de sequência e expressão gênica (KITANO, 2001).

A obtenção de dados experimentais de alto rendimento, coloca grandes demandas em processamento de bancos de dados, modelagem, simulação e medição de dados, tornando a biologia sistêmica um campo interdisciplinar, envolvendo engenharia, biologia molecular, ciência da computação, etc., também responsável por desenvolver tecnologias e ferramentas computacionais, onde a biologia dita qual nova tecnologia ou ferramenta computacional deve ser desenvolvida. Uma vez desenvolvida, essas ferramentas abrem novas fronteiras para a exploração dos fenômenos biológicos (ADEREM, 2005). Isso justifica a necessidade do estabelecimento de metodologias e técnicas por meio da investigação da estrutura dos sistemas, como genes, sistemas e transdução de sinais e estruturas físicas, dinâmicas de tais sistemas, métodos de controle de sistemas e métodos para projetar e modificar sistemas para propriedades desejadas (KITANO, 2001).

Ainda, segundo ADEREM (2005), existem três conceitos básicos que são essenciais para o entendimento de sistemas biológicos: (i) surgimento, (ii) robustez, e (iii) modularidade.

A respeito do surgimento, sistemas complexos apresentam propriedades emergentes que são demonstradas por suas partes individuais e não podem ser preditas, ainda que se conheça completamente as partes independentes. A vida é um exemplo de propriedade emergente. Não é inerente no DNA, RNA, proteínas, carboidratos ou lipídios, mas é consequência de suas ações e interações. O entendimento dessas propriedades requer perspectivas de nível de sistemas e não pode ser obtido em abordagens reducionistas

simples.

Sobre robusteza, sistemas biológicos mantém estabilidade fenotípica frente à diversas perturbações impostas pelo evento, podendo tanto serem eventos estocásticos como variações genéticas. Essa robusteza geralmente surge através de *loops* de retroalimentação positivos e negativos, e outras formas de controle que restringem a saída de um gene. Essa retroalimentação isola o sistema de flutuações impostos para o ambiente. O positivo, em geral, aumenta a sensibilidade, enquanto o negativo pode amortecer o ruído e rejeitar as perturbações. A robustez é uma propriedade inerente a todos os sitemas biológicos e é fortemente favorecida pela evolução (ADEREM, 2005).

Por fim, modularidade diz respeito à existência de unidades funcionais que interagem juntas para realizar uma função distinta. O módulo deve ter entradas e saídas. Para a biologia, um módulo em uma rede é um conjunto de nós com recursos e funções. A modularidade pode contribuir tanto para a robustez do sistema inteiro, limitando os danos a partes separáveis, e para evolução, simplesmente religando os módulos. Além disso, a modularidade diminui o risco de falha do sistema de prevenção ou espalha o dano em uma parte da rede por toda a rede.

Além disso, segundo KITANO (2001), para que um sistema biológico seja entendido enquanto sistema, é necessário que as estruturas do sistema sejam identificadas, primeiramente tais como relações regulatórias de genes e interações de proteínas que proverão transdução de sinal, vias de metabolismo, bem como as estruturas físicas do organismo, células, organelas, cromatina e outros componentes. Ambas relações topológicas da rede de componentes, bem como os parâmetros para cada relação precisam ser identificados. Ainda, métodos para identificar genes e redes metabólicas a partir de dados obtidos pelas técnicas de perfilamento de expressão gênica ainda precisam ser estabelecidos (KITANO, 2001). Após isso, é necessário entender o comportamento do sistema. Para isso, diversos métodos analíticos podem ser usados. Por exemplo, se o objetivo é determinar a sensitividade de certos comportamentos contra perturbações externas, e o quão rápido o sistema retorna ao seu estado normal após o estímulo, o estudo não só revela características em nível de sistema, mas também provê *insights* importantes para tratamentos médicos através da descoberta de resposta de células à certos componentes químicos. Em conjunto a isso, estabelecer um método para controlar o estado de sistemas biológicos é necessário. Esse método geralmente responde perguntas como: “Como transformar células malfuncionais em células saudáveis?”, “É possível controlar o estado da diferenciação para uma célula específica?”, entre outras. Por fim, é interessante estabelecer tecnologias que permitam projetar sistemas biológicos com o objetivo de prover curas para doenças.

2.3.1 Desafios Técnicos para Biologia Sistêmica

Grandes quantidades de genes e funções de seus produtos transcricionais têm sido identificados com o acompanhamento simbólico do sequenciamento completo do DNA. Sequências de DNA tem sido completamente identificados para vários organismos (KITANO, 2001). Métodos para perfilamento da expressão gênica estão disponíveis e proveem uma medida compreensiva no nível de mRNA. Além disso, diversos métodos têm sido inventados para introduzir perturbações na transcrição de genes, tais como o nocaute por perda de função de genes específicos e a interferência por RNA (RNAi).

Todos esses dados obtidos precisam ter uma padronização e um controle de qualidade para que seja utilizados em simulações e identificação de sistemas. As abordagens sistêmicas dependem fortemente da informação nos bancos de dados públicos. Esses bancos de dados são comumente incompletos, não padronizados ou devidamente anotados. Além disso, a qualidade dos dados é comumente incerta, o que reforça a necessidade de desenvolvimento de métricas para a validação de grandes conjuntos de dados (CHEN; MAR, 2018; PRATAPA *et al.*, 2020).

Para as abordagens computacionais serem bem sucedidas, as medidas devem ser abrangentes, detalhadas e sistemáticas. Mais especificamente sobre a abrangência, três pontos são importantes (KITANO, 2001): (i) abrangência do fator, (ii) abrangência de séries temporais, e (iii) abrangência de item. A abrangência do fator diz respeito ao número de genes e proteínas envolvidos. É importante que a medida seja realizada intensivamente para os fatores que estão relacionados aos genes centrais e proteínas de interesse. Em relação à abrangência de séries temporais, experimentos biológicos tradicionais tendem a medir apenas a mudança antes e após um certo evento. Entretanto, para métodos computacionais, dados médios em intervalos de tempo constantes são importantes. Por fim, a abrangência de itens envolve os níveis de diversidade, interação proteica, localização e outras características de medidas para um alvo específico. Isso dá-se devido ao fato da necessidade de obter integração entre todas as medidas e justifica a palavra sistêmica. Esse último ponto é especialmente importante para redes de regulação gênica pois dados de expressão podem ser inutilizáveis caso apenas o tipo selvagem seja medido. Desta maneira, os dados devem ter um conjunto abrangente de mutantes de deleção e superexpressão de cada gene.

Ainda, segundo (KITANO, 2001), necessidades futuras no desenvolvimento de redes biológicas vão desde o desenvolvimento de novos métodos teóricos para caracterizar a topologia da rede, até *insights* sobre a dinâmica de agrupamento de motivos e funções biológicas.

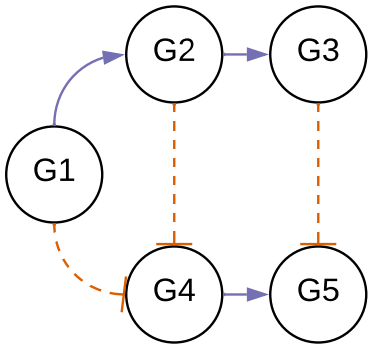
2.3.2 Redes de Regulação Gênica

Como discutido anteriormente, o genoma codifica milhares de genes cujos produtos permitem a sobrevivência celular e inúmeras funções celulares. As quantidades e o padrão temporal em que esses produtos aparecem na célula são cruciais para os processos da vida. As redes de regulação gênica controlam os níveis desses produtos gênicos. Elucidar as relações entre genes e os produtos que eles codificam permanece como um dos desafios centrais da biologia experimental e computacional (JACKSON *et al.*, 2020). Uma rede de regulação gênica é a coleção de espécies moleculares e suas interações, que juntas controlam a abundância de produtos genéticos. Numerosos processos celulares são afetados por redes regulatórias (KARLEBACH; SHAMIR, 2008).

Em uma rede de regulação gênica, genes são os nós e arestas representam as relações regulatórias (MCCALL, 2013). Uma rede contendo cinco genes (G1-G5) é apresentada na Figura 14. As setas azuis indicam relação de ativação e as barras vermelhas, inibição. Esta representação será utilizada ao longo deste trabalho. Ainda, a partir da Figura 14, é possível extrair as seguintes informações:

- G1 ativa G2 e inibe G4;
- G2 ativa G3 e inibe G4;
- G3 inibe G5;
- G4 ativa G5.

Figura 14 – Rede de Regulação Gênica com 5 genes (G1-G5). Setas azuis cheias indicam ativação e barras vermelhas tracejadas, inibição.



Fonte: Elaborado pelo autor (2022).

Diversas tentativas têm sido feitas para identificar redes de regulação gênica a partir de dados experimentais. Elas podem ser classificadas em duas abordagens: (i) *bottom-up* e, (ii) *top-down* (KARLEBACH; SHAMIR, 2008).

Em abordagens *bottom-up* o objetivo é construir uma rede de regulação gênica baseada na compilação independente de dados experimentais, em sua maioria através

de pesquisas na literatura e alguns experimentos específicos para obter dados sob vários aspectos da rede de interesse. Algumas das primeiras tentativas dessa abordagem são vistas no *lambda phase decision circuit* (MCADAMS; SHAPIRO, 1995), *early embryogenesis of Drosophila* (REINITZ; MJOLSNESS; SHARP, 1995; HAMAHASHI; KITANO, 1998; KITANO *et al.*, 1997), entre outras. Essa abordagem é adequada quando a maior parte dos genes e suas relações regulatórias são relativamente bem entendidas, onde a maior parte das peças do sistema é conhecida. Enquanto a maior parte dos parâmetros está disponível, o propósito principal é construir um modelo de simulação preciso que pode ser usado para analisar propriedades dinâmicas do sistema através da mudança de seus parâmetros que não podem ser feitas no sistema a fim de confirmar que o conhecimento disponível gera resultados de simulação que são consistentes com os dados experimentais disponíveis. Da mesma forma, essa abordagem pode ser aplicada em vias metabólicas, como o KEGG (KANEHISA; GOTO, 2000) e o EcoCyc (KARP *et al.*, 1999). Tais conjuntos de dados são extremamente úteis para modelagem e simulação.

Já a abordagem *top-down* tenta usar os dados de perfilamento de expressão gênica, seja em *microarrays* ou qualquer outra tecnologia de perfilamento. Alguns dos primeiros trabalhos utilizando técnicas de *clustering* são o ciclo celular da levedura (FEREA *et al.*, 1999; LASHKARI *et al.*, 1997; SPELLMAN *et al.*, 1998) e o desenvolvimento do sistema neural central de camundongos (CAMBRON *et al.*, 2012). Métodos de *clustering* são adequados para lidar com dados de expressão gênica em grande escala, mas não para deduzir diretamente as estruturas das redes. Tais métodos provêm apenas *clusters* de genes que estão co-expressados em padrões temporais similares. Muitas vezes, é necessária uma visualização de fácil compreensão (MICHAELS *et al.*, 1998). Além disso, heurísticas foram aplicadas para a inferência de redes neste tipo de abordagem.

Apesar de haver essas duas abordagens, KARLEBACH; SHAMIR (2008) ressaltam que é importante o desenvolvimento de um método híbrido que combine *bottom-up* e *top-down* pois é improvável que nenhum conhecimento esteja disponível antes de aplicar qualquer método de inferência. Em casos práticos, pode-se assumir que vários genes e suas interações são parcialmente entendidas, e que isso é necessário para identificar o resto da rede. Ainda, segundo (KARLEBACH; SHAMIR, 2008), uma pesquisa futura inclui a integração de dados de perfis de expressão, interações proteína-proteína e outros dados experimentais.

2.4 CIRCUITOS LÓGICOS

Todo circuito que obedece a um conjunto de regras lógicas pode ser denominado circuito lógico e escrito na forma de uma expressão lógica cuja função é descrever o relacionamento entre suas entradas e saídas utilizando a álgebra booleana (TOCCI; WIDMER; MOSS, 2003). A principal diferença entre a álgebra booleana e a convencional

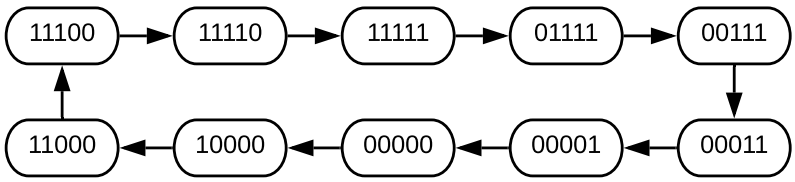
é que as variáveis podem assumir somente dois valores, 0 ou 1, que, no caso de circuitos digitais, representam os níveis de tensão que são aplicados.

Os circuitos lógicos podem ser classificados de duas maneiras distintas no que diz respeito ao seu comportamento em relação às suas entradas e suas saídas (TOCCI; WIDMER; MOSS, 2003):

- Circuitos Lógicos Combinacionais: são aqueles cujas saídas, em qualquer instante de tempo, dependem somente das combinações de suas entradas; e
- Circuitos Lógicos Sequenciais: são aqueles cujas saídas, além das combinações de suas entradas, dependem do estado anterior de elementos do próprio circuito. Os circuitos lógicos sequenciais podem ser entendidos também como circuitos lógicos combinacionais adicionados de elementos de memória, tais como *latches* e *flip-flops*.

A descrição das saídas de um circuito lógico pode ser feita através de uma tabela verdade. Esta tabela relaciona cada possível combinação de nível lógico presente nas entradas do circuito com o nível lógico das saídas. Outra representação é através de diagramas de transição de estados (TSD - do ingles *transition state diagram*) que exibem a sequência na qual os estados são atualizados. A Figura 15 apresenta um exemplo de um TSD de 5 variáveis, cíclico, onde o último estado do sistema retorna ao estado inicial.

Figura 15 – Diagrama de transição de estados. Considerando que o estado inicial é 11100, o TSD apresenta a evolução dos estados lógicos. Desta forma, o estado 11100 tem sua transição para o estado 11110, a transição seguinte é para o estado 11111 e assim sucessivamente.

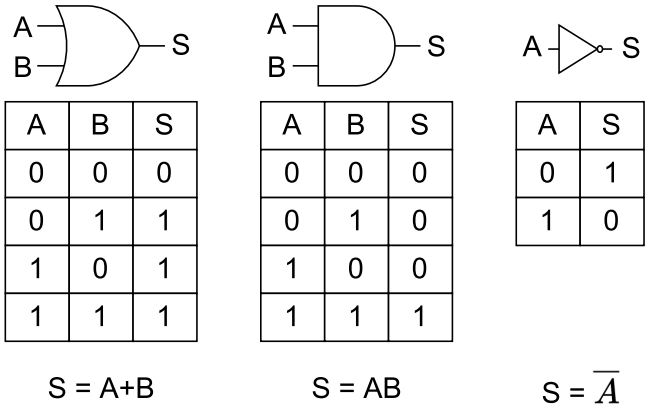


Fonte: Elaborado pelo autor (2022).

Qualquer circuito lógico pode ser expresso através das três operações lógicas básicas: OR (soma), AND (produto) e NOT (negação) TOCCI; WIDMER; MOSS (2003). Essas três operações são apresentadas na Figura 16 com suas respectivas simbologia, expressão e tabela verdade.

A expressão lógica pode ser extraída de uma tabela verdade utilizando-se de duas representações: soma de produtos e produto das somas. A primeira representação é mais comumente utilizada e pode-se obtê-la da seguinte maneira: cada linha da tabela verdade cuja saída está em nível lógico 1 é representada como o produto das suas entradas, denominado mintermo. Se a entrada está em nível lógico 1, utiliza-se a própria variável.

Figura 16 – As três portas lógicas básicas, suas simbologias, expressão lógica e tabela verdade.



A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

A	S
0	1
1	0

S = A+B S = AB S = $\overline{A}$

Fonte: Elaborado pelo autor (2022).

Caso a entrada esteja em nível lógico 0, nega-se a entrada. A expressão final é obtida somando-se todos os produtos.

A Tabela 1 exemplifica uma tabela verdade de 4 entradas ($ABCD$) e uma saída (F). Sua expressão lógica correspondente, na forma de soma de produtos, é apresentada na Equação 2.1.

Tabela 1 – Tabela verdade de 4 entradas e uma saída.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Fonte: Elaborado pelo autor (2022).

$$F = \overline{A}\ \overline{B}\ C\ \overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}D + ABCD \qquad (2.1)$$

No entanto, a maioria das expressões obtidas pela soma de produtos, como a apresentada na Equação 2.1 não estão na sua forma mais reduzida. Para este fim, diversas técnicas foram desenvolvidas a fim de simplificar a expressão e reduzir o número de elementos lógicos necessários para implementar as funcionalidades das expressões, tais como o mapa de Veitch-Karnaugh (TOCCI; WIDMER; MOSS, 2003) e ferramentas computacionais como o método de Quine Mc-Cluskey (MCCLUSKEY, 1956) e o ESPRESSO (BRAYTON *et al.*, 1984).

Os mapas de Veitch-Karnaugh (mapas K) utilizam como princípio a identificação visual de grupos de mintermos que podem ser simplificados. Os quadrados do mapa K são nomeados de modo que os quadrados adjacentes, tanto horizontalmente quanto verticalmente, difiram apenas em uma variável. Para utilizar o mapa K, basta dispor os minternos nas posições do mapa, exatamente como apresentado na tabela verdade e agrupar as regiões que apresentam nível lógico 1 no maior número possível de pares. Além disso, existe a condição de irrelevância (*don't care*, denotado por "X"), indicando que o nível lógico de uma saída não importa. Estas situações de irrelevância podem ser utilizados na formação de pares de simplificação. A simplificação é feita considerando-se os agrupamentos e analisando-se a mudança do nível lógico das variáveis. Se em um agrupamento uma variável é apresentada tanto no nível lógico 0 quanto no nível lógico 1, ela pode ser removida da expressão. O mapa K da Tabela 1 e o circuito resultante da simplificação são apresentados na Figura 17.

O método de Quine Mc-Cluskey (MCCLUSKEY, 1956), também chamado de método tabular, busca pelos agrupamentos em diversos níveis de acordo com o número de variáveis. Entretanto, este método possui alto custo computacional que aumenta exponencialmente com o número de variáveis (MANFRINI *et al.*, 2017).

A ferramenta computacional mais utilizada é o ESPRESSO (BRAYTON *et al.*, 1984) e seu funcionamento é baseado em uma heurística de minimização lógica. Ao invés de expandir uma função lógica em seus mintermos, o ESPRESSO manipula cubos, representando os termos de produto através de suas representações internas baseadas nos conjuntos de mintermos e *don't cares* iterativamente. Apesar do resultado da minimização não garantir o mínimo global, na prática ele é uma boa aproximação enquanto a solução é sempre livre de redundância quando são consideradas apenas as operações básicas (AND, OR e NOT). Na prática, o ESPRESSO não é capaz de lidar com simplificações entre as demais portas lógicas. Comparado a outros métodos, o ESPRESSO é essencialmente mais eficiente reduzindo o uso de memória e o tempo de computação em várias ordens de magnitude (BRAYTON *et al.*, 1984). Além disso, não existe restrição ao número de variáveis, funções de saída e termos de produto de um bloco de função combinacional. A entrada para o ESPRESSO é uma tabela verdade da funcionalidade desejada. O resultado é uma tabela reduzida descrevendo a função de acordo com as opções selecionadas. Por padrão, os termos de produto serão compartilhados o máximo possível pelas diversas

Figura 17 – Simplificação utilizando mapas K. O primeiro mapa (F) representa o posicionamento de todos os mintermos da função original. Os mapas 1, 2 e 3 são os agrupamentos dos mintermos para a simplificação. As simplificações são apresentadas nas expressões numeradas e o resultado da simplificação é apresentado na expressão F_s.

$$F = \overline{A}\,\overline{B}\,\underline{C}\,\overline{D} + \overline{A}B\underline{C}D + \overline{A}B\underline{C}D + \overline{A}B\underline{C}\overline{D} + \overline{A}B\underline{C}D + \overline{A}B\underline{C}D + \overline{A}B\underline{C}D$$

F		AB			
		00	01	11	10
CD	00				
	01		1	1	
	11		1	1	1
	10	1			1

1		AB			
		00	01	11	10
CD	00				
	01		1	1	
	11		1	1	1
	10	1			1

2		AB			
		00	01	11	10
CD	00				
	01		1	1	
	11		1	1	1
	10	1			1

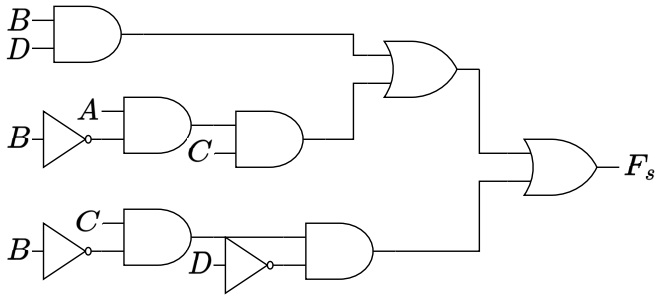
3		AB			
		00	01	11	10
CD	00				
	01		1	1	
	11		1	1	1
	10	1			1

1) $\overline{A}B\overline{C}D + AB\overline{C}D + \overline{A}BCD + ABCD = BD$

2) $A\overline{B}CD + A\overline{B}C\overline{D} = A\overline{B}C$

3) $\overline{A}\overline{B}CD + \overline{A}BC\overline{D} = \overline{B}C\overline{D}$

$F_s = BD + \overline{A}\overline{B}C + \overline{B}C\overline{D}$



Fonte: Elaborado pelo autor (2022).

funções de saída mas o programa pode ser instruído a lidar com as expressões de cada saída separadamente. Isso permite uma implementação eficiente em *arrays* lógicos de dois níveis como PLAs (*Programmable Logic Array*). O algoritmo do ESPRESSO provou-se tão bem sucedido que é incorporado como função de minimização lógica padrão em diversas ferramentas de síntese (MANFRINI *et al.*, 2017).

2.5 INFERÊNCIA E SOLUÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

Um conjunto de equações diferenciais ordinárias (EDOs) é uma forma de descrição matemática de um sistema físico ou biológico. Eles podem descrever as derivadas temporais de posições físicas ou concentrações químicas em função de seu estado atual (SCHMIDT; LIPSON, 2008). Encontrar uma forma simbólica para f(x,y) tal que a solução de um sistema composto por w equações diferenciais y = f(x, y) corresponda aos dados fornecidos é um problema de modelagem relevante (BERNARDINO; BARBOSA, 2010).

Dada uma classe de funções e um conjunto de observações experimentais, o problema de encontrar o elemento nesta classe que melhor se adequa aos dados é conhecido como regressão. Uma certa medida da distância entre a resposta prevista pela função/modelo e os dados disponíveis são então minimizados por um procedimento de otimização adequado. Em sua forma mais usual, a estrutura da função é predefinida e o problema é determinar certos coeficientes desta função. Entretanto, quando a estrutura da função não é definida *a priori*, tanto a estrutura quanto todos os parâmetros devem ser determinados. A isso dá-se o nome de regressão simbólica (AUGUSTO; BARBOSA, 2000).

O objetivo é, portanto, encontrar o sistema de equações diferenciais $y' = f(x, y) = \{y_1^{'} = f_1(x, y), y_2^{'} = f_2(x, y), \ldots, y_n^{'} = f_n(x, y)\}$ que descrevem o comportamento dos pares observados (x_k, y_k), onde $y_k = y(x_k)$, $k = 1, ..., m$ e m é o tamanho do conjunto de dados.

Existem pelo menos duas maneiras de resolver esse problema usando técnicas de regressão simbólica. A primeira consiste em derivar numericamente os dados, obtendo um conjunto de aproximações $\overline{y}_k^{'} \approx y_k^{'}$. Isso pode ser feito, por exemplo, utilizando a diferenciação numérica de 2 pontos, onde:

$$\overline{y}_k^{'} = \frac{y_{k+1} - y_k}{h}, \tag{2.2}$$

Outra alternativa é integrar numericamente o sistema (solução candidata) e comparar o resultado com os pares de dados observados (x_k, y_k). O método *Runge-Kutta* de 4ª ordem pode ser utilizado nesse caso:

$$\begin{aligned} \overline{y}_{k+1} &= \overline{y}_k + \tfrac{h}{6}(q_1 + 2q_2 + 2q_3 + q_4) \\ q_1 &= f(t_k, \overline{y}_k) \\ q_2 &= f(t_k + \tfrac{h}{2}, \overline{y}_k + \tfrac{h}{2}q_1) \\ q_3 &= f(t_k + \tfrac{h}{2}, \overline{y}_k + \tfrac{h}{2}q_2) \\ q_4 &= f(t_k + h, \overline{y}_k + hq_3) \end{aligned} \tag{2.3}$$

onde $\overline{y}_0 = y_0$ é o valor inicial.

Esta segunda abordagem tende a ser mais precisa, pois o método Runge-Kutta de 4ª ordem tem um erro da ordem h^5 enquanto o erro usando a diferenciação numérica de dois pontos é da ordem h, onde h é o tamanho do passo. No entanto, também é mais caro

computacionalmente porque é necessária uma integração numérica para avaliar o erro de cada solução candidata (sua afinidade) enquanto que a diferenciação numérica requer que a diferenciação ocorra apenas uma vez no início do processo de busca (BERNARDINO; BARBOSA, 2010).

2.6 PROBLEMA DE OTIMIZAÇÃO

Um problema de otimização é aquele onde se procura determinar os valores extremos de uma função, isto é, o maior ou o menor valor que uma função pode assumir em um dado intervalo (GIBIM, 2015). A forma padrão de um problema de otimização mono-objetivo é dado por

$$\begin{aligned} &\text{Otimizar} && \mathbf{f(x)} \\ &\text{sujeito a} && \\ & && g_i(x) \leq 0,\ i = 1, ..., p \\ & && h_j(x) = 0,\ j = 1, ..., q \end{aligned} \tag{2.4}$$

onde $f(x)$ é a função objetivo a ser maximizada ou minimizada sobre a variável x, $x \in \mathbb{R}^n$. $g_i(x) \leq 0$ é chamada de restrições de desigualdade, e $h_j(x) = 0$ é chamada de restrições de igualdade.

2.7 COMPUTAÇÃO EVOLUCIONISTA

Um grande marco na biologia foi a publicação do livro "A origem das espécies" (DARWIN, 1969), onde pela primeira vez relacionava-se a sobrevivência das espécies com o ambiente no qual elas estavam inseridas. Darwin concluiu que os mais adaptados ao ambiente tendem a sobreviver por mais tempo e, por conseguinte, geram mais descendentes. Essa conclusão era reforçada pelos mecanismos genéticos, onde as características parentais se misturam no organismo da prole, que haviam sido descobertos pouco tempo antes da proposta de Darwin (MICHALEWICZ, 1996). Esses conceitos serviram de inspiração para o desenvolvimento da computação evolucionista. Desta forma, a computação evolucionista constitui uma área de pesquisa de ciência da computação que simplifica as questões de teoria da evolução e genética a fim de resolver problemas complexos de otimização.

Nas próximas seções são discutidos os princípios empregados nos algoritmos evolutivos que servem como base para apresentação das Estratégias Evolutivas e Programação Genética Cartesiana, tendo em vista que essas são as metaheurísticas utilizadas neste trabalho.

2.7.1 Algoritmos Evolutivos

A ideia subjacente comum entre a maioria dos algoritmos evolutivos (AEs) é basicamente a mesma: dada uma população de indivíduos e um ambiente que possui

recursos limitados, a competição por esses recursos causa seleção natural (sobrevivência dos mais aptos). Isso, por sua vez, contribui para o aumento na aptidão dos indivíduos dessa população. Dada uma função de qualidade a ser maximizada, pode-se criar aleatoriamente um conjunto de soluções candidatas (EIBEN; SMITH *et al.*, 2003). Com base nos valores de aptidão, alguns dos melhores candidatos são escolhidos para semear a próxima geração. Isso é feito através da aplicação de operadores de variação (recombinação e/ou mutação). Enquanto a recombinação é um operador que é aplicado a dois ou mais candidatos selecionados (pais), produzindo um ou mais candidatos (filhos), a mutação é aplicada a um candidato e resulta em um novo candidato. Os novos candidatos são avaliados e, em seguida, competem com os demais indivíduos da população com base em sua aptidão. Este processo pode então ser repetido até que um candidato com qualidade suficiente (solução) seja encontrado ou até que algum critério de parada pré-estabelecido seja atingido. Portanto, o processo evolutivo resulta em uma população que está cada vez mais adaptada ao meio ambiente. Estes conceitos servem como base para diversos algoritmos, tais como as Estratégias Evolutivas e a Programação Genética Cartesiana, utilizadas neste trabalho, e discutidos nas próximas seções.

2.7.2 Estratégias Evolutivas (ES)

As Estratégias Evolutivas (ES - do inglês *Evolution Strategies*) (BACK, 1996; ROZENBERG; BÄCK; KOK, 2012) podem ser descritas como uma especialização de algoritmos evolutivos (EIBEN; SMITH *et al.*, 2003), e são caracterizadas por quatro propriedades:

1. A seleção de indivíduos para recombinação não tem viés,
2. A seleção para substituição da população é um processo determinístico,
3. Operadores de mutação são parametrizados e podem mudar suas propriedades durante a otimização, e
4. Indivíduos consistem de parâmetros de decisão e, em algumas variantes, parâmetros de estratégia.

O procedimento geral da ES é apresentado no Algoritmo 1, onde o operador de variação é definido no conjunto de parâmetros Ψ_V e o operador de avaliação é explicitamente mencionado. Uma população na geração t ≥ 0 é denotado por $P^{(t)}$ e é um conjunto de indivíduos. Um indivíduo $p \in P^{(t)}$ é uma tupla (x, Ψ). Os conjuntos Ψ e Ψ_V são conjuntos finitos arbitrários que representam os parâmetros de estratégia. Pelo fato dos parâmetros de estratégia serem modificados internamente durante a execução do algoritmo, são chamados de parâmetros de estratégia endógenos. O número de indivíduos pai é denotado por μ e o número de descendências por λ. Além disso, ρ denota o número de pais que são

considerados para gerar uma única descendência através de recombinação. Para estes parâmetros, $\mu, \rho, \lambda \in \mathbb{N}$ e $\rho \leq \mu$.

Algoritmo 1: Procedimento geral das estratégias evolutivas (EIBEN; SMITH *et al.*, 2003).

Inicialização de $P^{(0)}$ com μ indivíduos
t $\leftarrow 0$
repeat
$Q^{(t)} \leftarrow \emptyset$
for $i = 1 \rightarrow \lambda$ **do**
  amostra ρ pais $p_1, \ldots, p_\rho \in P^{(t)}$ uniformemente de maneira aleatória
  q $\leftarrow$ variação$(p_1, \ldots, p_\rho, \Psi_V)$
  $Q^{(t)} \leftarrow Q^{(t)} \cup \{q\}$
end for
$P^{(t+1)} \leftarrow$ seleção dos μ melhores indivíduos de $Q^{(t)} \cup \{p \in P^{(t)} : \text{idade} < \kappa\}$
Atualiza Ψ_V
$\forall p \in P^{(t+1)}$: atualiza idade
$t \leftarrow t + 1$
até que o critério de parada seja atingido

Ainda, $\kappa \in \mathbb{N}$ representa a maior idade que pode ser atingida por um indivíduo na população. Ao contrário dos parâmetros endógenos, μ, ρ, λ e κ devem ser determinados pelo usuário. Por esse motivo são denominados parâmetros de estratégia exógenos. A configuração de κ tem um impacto direto no operador de seleção. Usualmente, $\kappa = 1$ (uma geração de vida) ou $\kappa = \infty$ (sem limite de tempo de vida) são usados. Para $\kappa = 1$, denomina-se seleção vírgula e para $\kappa = \infty$, seleção adição, representados por $(\mu/\rho, \lambda)$-ES e $(\mu/\rho + \lambda)$-ES, respectivamente. A seleção adição é elitista e comumente opta-se pelo uso da seleção vírgula pois, segundo (EIBEN; SMITH *et al.*, 2003):

- Há maior facilidade de escapar de ótimos locais,
- Tende a ser melhor em seguir um ótimo que se move, e
- O uso da seleção adição permite que valores ruins de parâmetros de mutação sobrevivam por muito tempo caso o progenitor esteja bem adaptado.

Como mencionado anteriormente, uma grande e importante característica da ES é a adaptabilidade do parâmetro de mutação. Três requerimentos básicos dos operadores de mutação são (EIBEN; SMITH *et al.*, 2003): (i) todo ponto no espaço de busca precisa ser alcançável com uma probabilidade estritamente maior que zero pela aplicação de um número finito de mutações, (ii) deve ser não enviesada, e (iii) ser parametrizado. Esses requerimentos são satisfeitos por uma distribuição normal multivariada. Desta forma, a mutação recebe a notação N(ζ, σ), onde ζ (média) = 0 e σ é o desvio padrão. Um outro fato a ser considerado é que a ordem da mutação é importante. Seja uma mutação <x, σ>

$\rightarrow$ <x', σ'>, primeiro muta-se $\sigma \rightarrow \sigma'$ e, então, x $\rightarrow$ x' = x + N(0, σ'). Desta forma, x' é bom se f(x') é bom e σ' é bom se x' criado a partir de σ' é bom, sendo f(x) a função de avaliação (BÄCK; FOUSSETTE; KRAUSE, 2013).

A primeira ES, conhecida como (1+1)-ES, gera uma única descendência x' a partir de um único progenitor ($x \in \mathbb{R}$). O valor de x' era atualizado segundo x' = x + $\sigma.N(0,1)$. Desta forma, x' se torna pai se f(x') > f(x), quando objetiva-se maximizar. Se o problema é de minimização, x' torna-se pai se f(x') < f(x). Além disso, o parâmetro de tamanho de passo de mutação σ é fixo. Por passo de mutação, entende-se a força da mutação, onde maiores valores indicam maior probabilidade de ocorrência de mutações. Posteriormente adicionou-se a adaptação de tamanho de passo da mutação, utilizando a regra do 1/5 de sucesso. Esta regra diz que se as mutações são bem sucedidas (geram indivíduos melhores que seus pais) em torno de 1/5, não há necessidade de adaptação de passo. Se a taxa de sucesso é inferior a 1/5 reduz-se o tamanho do passo e aumenta-se o tamanho do passo caso contrário. Desta forma, $\sigma' = \sigma \ . \ c^{\{-1,\ 1\}}$, com 0,817 $\leq$ c $\leq$ 1, onde c = 0,817 é uma constante derivada por Schwefel (SCHWEFEL, 1977). Desde então, diversas outras variantes de ES surgiram (BÄCK; FOUSSETTE; KRAUSE, 2013).

2.7.3 Programação Genética Cartesiana (CGP)

A programação genética (GP - do inglês *genetic programming*) é um membro da família dos algoritmos evolutivos (EIBEN; SMITH *et al.*, 2003). Sua diferença para os demais algoritmos evolutivos não reside somente na aplicação mas também em sua representação, onde, diferentemente dos algoritmos genéticos, árvores foram amplamente adotadas como cromossomos (EIBEN; SMITH *et al.*, 2003), principalmente por conta dos trabalhos de Koza (KOZA, 1994). O paradigma da programação genética continua a tendência de lidar com o problemas antes abordados pelos algoritmos genéticos, mas aumentando a complexidade das estruturas. Em particular, as estruturas que sofrem adaptação na programação genética são gerais, com tamanho e forma dinamicamente variados (KOZA, 1994). Em geral, na programação genética tradicional, populações de centenas ou milhares de soluções candidatas são criadas.

A Programação Genética Cartesiana (CGP) (MILLER, 2011) é uma técnica de programação genética para a evolução automática de programas de computador e outras estruturas computacionais. Originalmente, a CGP foi desenvolvida para a evolução de circuitos digitais (MILLER; THOMSON; FOGARTY, 1997). Atualmente, é possível encontrar a aplicação da CGP em diversas áreas, tais como controladores robóticos (GARCÍA; COELLO, 2002), redes neurais (GOLDMAN; PUNCH, 2013) e classificadores de imagem (GOLDMAN; PUNCH, 2015).

Diferentemente da programação genética clássica, na CGP os indivíduos são representados como grafos direcionados acíclicos (DAGs - do inglês *Directed Acyclic Graphs*),

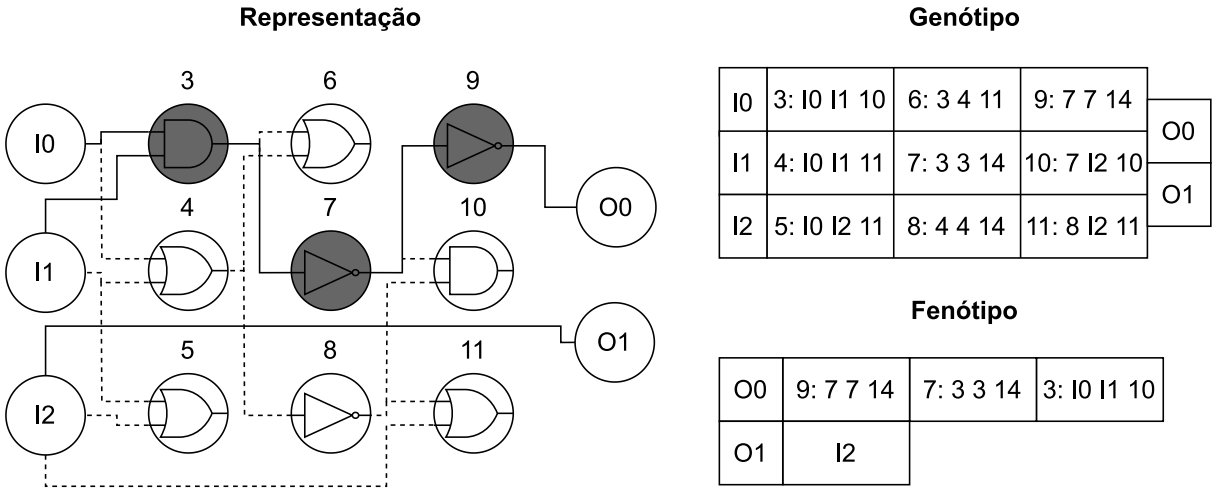
representados como uma matriz bidimensional de nós. O genótipo é composto por genes (inteiros) que representam de onde um nó obtém suas entradas (genes de conexão) e a operação realizada por este nó (genes de função). A quantidade de genes de conexão é denominada aridade do nó. A função que um nó realiza depende da aplicação. O conjunto de funções é representado por Γ e, no caso de circuitos digitais, é composto por funções booleanas, tais como $\Gamma = \{AND, OR, NOT, XOR\}$. Além disso, a CGP tem ao menos três parâmetros definidos pelo usuário: número de colunas (n_c), número de linhas (n_r), e o *levels-back*. Os dois primeiros parâmetros definem a topologia da matriz. Um caso especial desses três parâmetros ocorre quando o número de linhas é um e *levels-back* é igual ao número de colunas. Desta forma o genótipo pode representar qualquer grafo direcionado acíclico (MILLER, 2011), limitado ao número de elementos disponíveis. O parâmetro *levels-back* controla a conectividade do grafo, restringindo quais colunas (à esquerda do nó) um nó pode obter suas entradas. De qualquer forma, qualquer nó pode conectar-se diretamente às entradas primárias do circuito, independentemente do valor de *levels-back*. É importante ressaltar que este parâmetro é restringido ao intervalo [1, n_c], já que n_c é o número máximo de colunas. Além disso, caso considere-se que índices negativos representam entradas que vêm da direita, haveriam ciclos. Esse fato é especialmente importante para a evolução de circuitos lógicos combinacionais que, por definição, não podem possuir retroalimentação.

O genótipo da CGP é comumente decodificado percorrendo-se os nós desde a saída até as entradas primárias. Durante esse processo, os nós que se conectam às saídas, mesmo que indiretamente, são denominados nós ativos (fenótipo). Os nós que não estão presentes no fenótipo são denominados nós inativos. Um indivíduo da CGP com três entradas e duas saídas, sua matriz de representação (genótipo) e a representação de sua decodificação (fenótipo) são apresentados na Figura 18.

O genótipo é composto pelas três entradas primárias (I0, I1 e I2) seguido pelos nós 3 a 11. Já o fenótipo é composto pelas entradas primárias I0 e I1 e os nós 3, 7 e 9 para a saída O0 e pela entrada primária I2 para a saída O1. A quantidade de nós presentes no genótipo da CGP é fixo, e é determinado pelo produto do número de linhas pelo número de colunas ($L_n = n_c \times n_r$).

A presença de nós inativos é uma característica importante da CGP pois esses nós podem se tornar ativos a partir da aplicação de um operador genético. Outro fator importante é que, ainda que dois indivíduos da população tenham a mesma aptidão, é possível que seus nós inativos sejam diferentes. A presença de nós ativos auxilia o algoritmo a não ficar preso em ótimos locais e existe um amplo estudo sobre a importância deles (TURNER; MILLER, 2015), conhecido como Deriva Gênica Neutra (NGD - do inglês *Neutral Genetic Drift*). Os nós inativos na CGP, segundo MILLER (2011), correspondem a 95% de todo o genótipo.

Figura 18 – Indivíduo da CGP com 3 entradas (I0, I1 e I2) e 2 saídas (O0 e O1). Os nós em cinza são ativos, pois contribuem diretamente para a saída e os brancos, inativos. Linhas tracejadas representam conexões entre nós inativos e linhas contínuas definem o fenótipo.



Fonte: Elaborado pelo autor (2022).

A técnica de busca mais comum na CGP é uma Estratégia Evolutiva (1 + λ)-ES (MILLER, 2011), onde λ é o número de novas soluções geradas a cada iteração do algoritmo e comumente adota-se 4. Entretanto, é possível modificar não só a quantidade de filhos gerados como também o mecanismo de busca. Em SILVA *et al.* (2022), a (1 + λ)-ES é substituída por um CLONALG[4], por exemplo.

Quando utilizando uma (1 + λ)-ES, o melhor indivíduo entre o progenitor e as λ novas soluções geradas é selecionado para a próxima geração. Em caso de empate, seleciona-se um entre os melhores filhos por conta das questões de deriva gênica neutra. Esses λ novos indivíduos são criados a partir da clonagem do progenitor e a mutação destes clones. Um pseudocódigo da CGP é apresentado no Algoritmo 2.

Técnicas evolutivas devem estar equipadas com um procedimento adicional de tratamento de restrições quando resolvendo problemas de otimização com restrições (MEZURA-MONTES; COELLO, 2011). Esse é o caso do problema de otimização à respeito do projeto de circuitos lógicos combinacionais, e um método de tratamento de restrições deve ser adicionado à CGP. Desta forma, adota-se um esquema de seleção por torneio (TS - do inglês *tournament selection*), similar ao proposto em (DEB, 2000) e comumente adotado na literatura (SILVA; BERNARDINO, 2018; SILVA; SOUZA; BERNARDINO, 2019; MANFRINI; BERNARDINO; BARBOSA, 2016; MANFRINI *et al.*, 2017; MILLER,

[4] O algoritmo de seleção clonal (CLONALG) é uma abordagem de sistema imunológico artificial que envolve os anticorpos inspirados no conceito da expansão clonal e seleção. De acordo com esse método, cada célula (solução candidata) é clonada, hipermutada e aqueles com maior afinidade antigênica (qualidade) são selecionadas.

Algoritmo 2: CGP com $(1 + \lambda)$-ES. Adaptada de (MILLER, 2011).

```
μ ← 1 (número de progenitores)
λ ← 4 (numero de descendências)
Progenitor ← ∅
forall i tal que 0 ≤ i < μ + λ do
    Gera aleatoriamente um indivíduo i
end forall
Progenitor ← o melhor indivíduo
while critério de parada não é atingido do
    forall i such that 0 ≤ i < λ do
        i → Muta(Progenitor)
        Fenótipo[i] → DecodificaGenótipo(i)
        Avalia(i)
    end forall
    Progenitor ← o melhor indivíduo
end while
```

2011), onde o melhor indivíduo é obtido de acordo com os seguintes critérios:

1. qualquer solução factível é preferível em relação à uma solução infactível,
2. dentre duas soluções factíveis, aquela que tiver menor violação de restrições é preferível, e
3. dentre duas soluções infactíveis, aquela com menor violação de restrições é preferível.

Aqui, uma solução factível é aquela que atende integralmente a tabela verdade. O nível de violação de restrição é definido como $v(x) = \sum_{k=0}^{n_h} h_k(x)$, onde $h_k(x)$ representa as restrições. Isso é equivalente à *hamming distance* entre as saídas de um circuito candidato x e aquelas do circuito factível x_f. Além disso, uma nova solução (filho) é preferível em relação ao seu pai quando:

1. não existe melhoria no nível de violação e $v(x) > 0$, e
2. não observa-se melhora no valor da função objetivo de soluções factíveis.

Um dos melhores filhos é selecionado em caso de empate.

Neste trabalho, a CGP é utilizada como algoritmo de inferência de modelos booleanos de redes de regulação gênica.

2.7.3.1 Operadores de Variação

O principal operador de variação genética na CGP é a mutação (MILLER, 2011). Diversos operadores de mutação foram desenvolvidos para a CGP (GOLDMAN; PUNCH, 2013; SILVA; SOUZA; BERNARDINO, 2019; HODAN; MRAZEK; VASICEK, 2020).

Neste trabalho concentraremos as explicações em duas: *point mutation (PM)* e *single active mutation (SAM).*

A mutação pontual (PM - do inglês *point mutation*) é mais similar às mutações simples utilizadas em programação genética. Existe um parâmetro (μ_r) que define a porcentagem do genótipo que sofrerá mutação. O número de nós mutados é, portanto, definido como $L_n \times \mu_r$. Como o resultado dessa operação geralmente gera um número real, a quantidade de nós mutados pode ser tanto o *ceil* quanto o *floor* desse valor, a depender da implementação. Comumente opta-se pelo *ceil* para que haja garantia de mutação de ao menos um nó, caso o resultado seja um valor menor que 1 (SOUZA *et al.*, 2020; SILVA; BERNARDINO, 2018; SILVA; SOUZA; BERNARDINO, 2019). Esses nós são selecionados aleatoriamente e podem sofrer uma mutação tanto em seus genes de conexão quanto seus genes de função. Entretanto é possível que a mutação pontual modifique somente nós inativos, o que resulta em filhos fenotipicamente idênticos aos pais.

Tendo em vista essa questão, e o fato de que os filhos fenotipicamente idênticos aos seus pais são reavaliados após a mutação, GOLDMAN; PUNCH (2013) propuseram um operador denominado *single active mutation (SAM)* com o objetivo de reduzir o número de avaliações desnecessárias (filhos fenotipicamente idênticos aos pais), garantindo que todo filho sofra uma mutação em um nó ativo. O procedimento do SAM é simples e pode ser obtido de acordo com os seguintes passos: (i) seleciona-se aleatoriamente um nó do genótipo, e (ii) aplica-se a mutação. Os passos (i) e (ii) são repetidos até que um nó ativo tenha sido selecionado e mutado. Esse tipo de mutação tem sido aplicado com sucesso na evolução de circuitos lógicos combinacionais (SOUZA *et al.*, 2020; VASICEK, 2015; VASICEK; SEKANINA, 2015; VASICEK, 2018). Uma vantagem desse operador de mutação é a inexistência de parâmetros. Entretanto, como a maior parte dos nós de um indivíduo da CGP são inativos, não existe controle de quantas mutações são realizadas por geração.

Um outro operador de mutação utilizado neste trabalho é o operador de mutação orientado semanticamente (SOMO - do inglês semantically oriented mutation operator) (HODAN; MRAZEK; VASICEK, 2020), que foi desenvolvido para o projeto de circuitos digitais usando CGP, onde o operador de mutação puramente estocástico é substituído e opera no espaço fenotípico. O objetivo principal do SOMO é garantir que a mutação efetuada seja sempre a melhor possível para o nó selecionado. Os principais passos do SOMO são apresentados no Algoritmo 3 que pode ser resumido como: (i) uma quantidade definida (p_q) de nós inativos sofrem mutação em suas entradas e funções, (ii) um nó aleatório (c) é selecionado a fim de receber a mutação SOMO, (iii) c tem sua função modificada aleatoriamente com uma probabilidade p_f, (iv) uma entrada aleatória e é escolhida de c, e (v) a entrada aleatória e é conectada ao melhor nó possível. O passo (i) garante que novo material genético é gerado antes de realizar a mutação. Com a entrada aleatória e selecionada, SOMO identifica a melhor entrada considerando todos os nós

anteriores no genótipo e, então, realiza a mutação. A identificação do nó mais adequado

Algoritmo 3: Operador de Mutação Orientado Semanticamente (HODAN; MRAZEK; VASICEK, 2020)

Entrada: Um indivíduo da CGP p consistindo de $|C|$ nós;
Saída: Um indivíduo mutado p';
(C, E, C_{PI}, C_{PO}, Ψ) $\leftarrow$ decodifica(p) ; `/* decodifica p como um DAG (C, E) com` C_{PI} `folhas e` C_{PO} `raízes (saídas); Ψ: C → Γ */`
N $\leftarrow$ activeNodes; c $\leftarrow$ selectNodeRandomly(N / C_{PI});
if *(rand(0,1)* $< p_f \wedge$ *(*$c \notin C_{PO}$*)* **then**
  /* mutação da função */
  $\Psi(c) \leftarrow$ rand(0, Γ-1);
else
  /* mutação de entrada */
  muta entrada e função de p_q nós inativos;
  e $\leftarrow$ selectInputEdgeRandomly({(x, c) $\in$ E|x $\in$ N)};
  n $\leftarrow$ identifyBestNode(c, e, (C,E), Ψ); E $\leftarrow$ (E / {e}) $\cup$ {(n, c)} ;
end if
Retorna p' $\leftarrow$ encode(C,E,C_{PI},C_{PO},Ψ)

a ser conectado no nó c é baseado em semântica e apresentado no Algoritmo 4. Esse procedimento pode ser resumido como (i) calcula-se o *score* de cada nó do genótipo que pode ser conectado ao nó c, e (ii) se nós recebem o mesmo *score*, o nó mais próximo às entradas primárias é preferido.

O *score* reflete a *hamming distance*, que é a distância entre a especificação dada pela tabela verdade e a resposta do circuito candidato p. Na sequência, todos os nós à esquerda do nó a ser mutado c são conetados em e e simula-se de três maneiras:

1. usando a função do nó que está sendo conectado,
2. forçando e à nível lógico 0, e
3. forçando e à nível lógico 1 ($val_{e=1}^{[o]}$).

O valor de entrada desejado é denotado como *req* e pode ser igual a 0, 1 ou X, onde X é uma situação de irrelevância. O termo [o] em sobrescrito aponta para um valor Booleano associado a um nó de saída do programa o. Além disso, o valor de *req* é determinado utilizando o operador ternário Θ

$$\Theta(t, v_0, v_1) = \begin{cases} X, & v_0 = v_1 \\ 0, & v_0 = t \\ 1, & v_1 = t \end{cases} \tag{2.5}$$

e o operador de redução $\bigodot$, definido como

$$\bigodot(a,b) = \begin{cases} a, & a \neq X \\ b, & \textit{caso contrário.} \end{cases} \tag{2.6}$$

Algoritmo 4: Procedimento identifyBestNode (HODAN; MRAZEK; VASICEK, 2020).

Entrada: DAC(C,E), atribuição de função do nó Ψ, nó selecionado c e sua entrada e, especificação da tabela verdade *TT(x)*, onde x $\in \mathbb{B}^{n_i}$, $\mathbb{B} = \{$'0', '1'$\}$
Saída: O nó mais adequado $n \in$ C
inicializa score(n) de 0 para todos $n \in$ C
N $\leftarrow$ candidatos para conexão
forall $x \in \mathbb{B}^{n_i}$ **do**
  determinar o valor de entrada desejado para cada saída
  $val \leftarrow$ avalia N para entrada x;
  $val_{e='0'} \leftarrow$ avalia C / N para entrada x e e forçado a '0';
  $val_{e='1'} \leftarrow$ evaluate C / N para entrada x e e forçado a '1';
  $req \leftarrow \bigodot_{o \in C_{PO}} (\Theta(TT(x)^{[o]}, val_{e='0'}, val_{e='1'}))$
  forall $n \in N$ **do**
    atualiza o *score* de cada nó
    score(n) $\leftarrow$ score(n) + HD*(req, $val^{[n]}$);
  end forall
end forall
Retorna $argmax_{n \in N}$score(n)

Diferentemente da CGP tradicional, SOMO usa $\lambda = 1$ e a inicialização da população é feita com uma solução candidata não tendo nós ativos a fim de maximizar a eficiência e minimizar o número de nós ativos das soluções evoluídas (HODAN; MRAZEK; VASICEK, 2020). A inexistência de nós ativos nas soluções candidatas da população inicial indicam conexões diretas das saídas nas entradas. O próprio operador SOMO é responsável pela ativação dos nós durante o processo evolutivo.

Os operadores de recombinação não receberam atenção na CGP devido ao fato de que tentativas iniciais do uso deste operador mostraram-se prejudiciais aos subrafos da CGP (MILLER, 2011). Entretanto é possível encontrar operadores de recombinação que geram bons resultados quando aplicados a problemas específicos (WALKER; MILLER; CAVILL, 2006; CLEGG; WALKER; MILLER, 2007; KALKREUTH; RUDOLPH; DROSCHINSKY, 2017; HUSA; KALKREUTH, 2018; SILVA; BERNARDINO, 2018).

2.7.3.2 **Paralelismo**

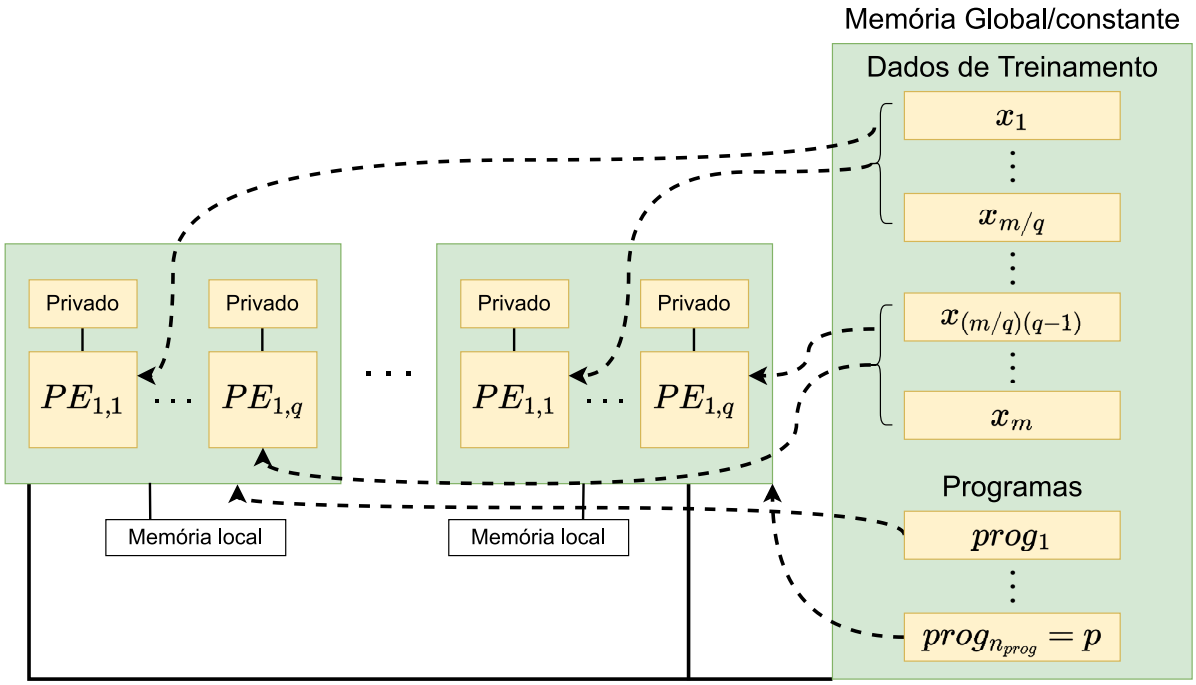
Algoritmos evolutivos são naturalmente paralelizáveis, de tal maneira que a população de soluções candidatas de cada iteração/geração pode ser avaliada em paralelo.

A parte mais cara computacionalmente de algoritmos evolutivos é a avaliação dos indivíduos (SUDHOLT, 2015). Além disso, é importante ressaltar que no caso da evolução de CLCs, o tamanho do problema cresce exponencialmente com o número de entradas.

KOZA (1992) propõe o uso de duas abordagens para lidar com a avaliação de técnicas de PG, quando se fala de problemas envolvendo uma base de dados, em paralelo: paralelizar a avaliação completa do indivíduo, de tal maneira que isso aconteça simultaneamente para todos os indivíduos ou, no nível de instâncias, onde cada indivíduo executa sequencialmente mas em paralelo sobre o conjunto de dados.

A estratégia adotada em AUGUSTO; BARBOSA (2013) e utilizada em SILVA; BERNARDINO; BARBOSA (2021) para CGP, envolve o paralelismo de população por unidade computacional e a avaliação de cada indivíduo é paralelizada explorando o paralelismo de dados nos elementos processantes, como apresentado na Figura 19.

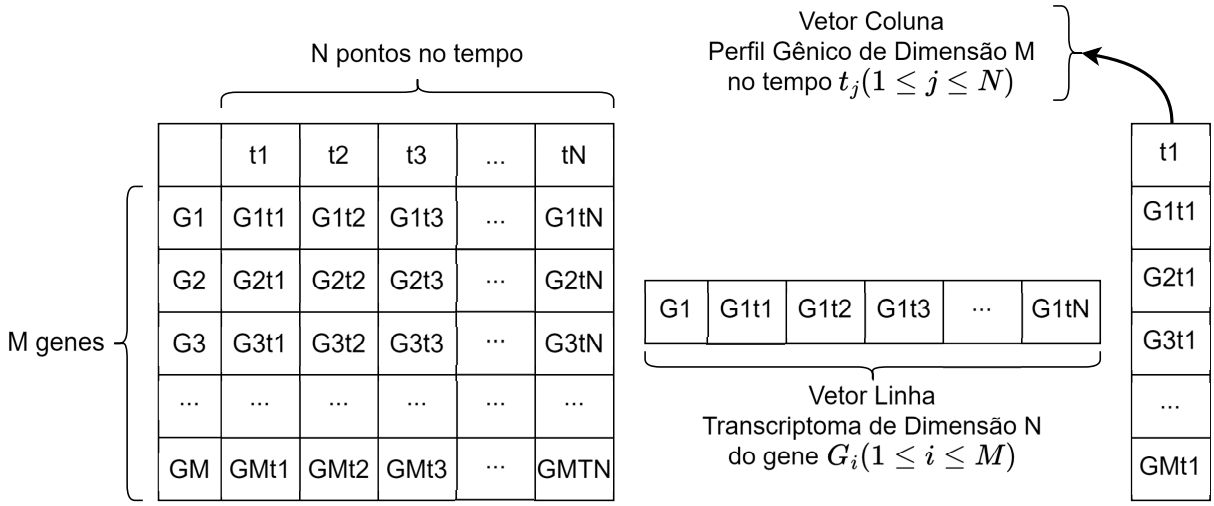
Figura 19 – Esquema de paralelismo usado no P-CGPANN (SILVA; BERNARDINO; BARBOSA, 2021)



Fonte: Adaptado de (AUGUSTO; BARBOSA, 2013) (2024).

A fim de paralelizar e acelerar a avaliação das soluções candidatas e, consequentemente, a evolução dos modelos, PRACHEDES *et al.* (2022a) utilizaram a mesma estratégia para a inferência de modelos de GRNs utilizando CGP com paralelismo em unidades de processamento gráfico (GPU - do inglês *graphics processing units*), através de OpenCL.

Figura 20 – Matriz de Expressão Gênica. Cada vetor linha representa um transcriptoma de dimensão N e cada vetor coluna corresponde a um perfil gênico de dimensão M.



Fonte: Elaborado pelo autor (2023).

2.8 CARACTERIZAÇÃO DO PROBLEMA

O problema principal abordado neste trabalho consiste em inferir a topologia e a intensidade das relações de uma rede de regulação gênica a partir de dados de expressão gênica. Uma expressão gênica é um conjunto de dados em uma matriz $M \times N$, onde cada vetor linha s_i (i = 1, ..., N) representa um transcriptoma de dimensão N, e cada vetor coluna y_j (j = 1, ..., M) corresponde a um perfil gênico de dimensão M, onde M e N são o número total de genes e os perfis gênicos, respectivamente (CHEN; MAR, 2018), conforme ilustrado na Figura 20.

O perfil gênico pode ser uma condição experimental, tal como a realização de diversas perturbações nos genes ou uma série temporal representando a variação da expressão gênica em pontos no tempo. Desta forma, cada coluna y_j é um ponto no tempo.

O objetivo do método de inferência da rede é usar essa matriz de dados para predizer um conjunto de relações regulatórias entre quaisquer dois genes de um total de M genes (AALTO *et al.*, 2020). A saída final é dada na forma de relações lógicas que servem como base para a construção de um grafo com M nós e um conjunto de arestas (CHEN; MAR, 2018). As arestas apresentam um peso que determina a intensidade da relação regulatória entre os dois nós em questão. Além disso, as relações lógicas obtidas são utilizadas para a obtenção de um modelo contínuo na forma de um sistema de EDOs, através do qual é possível reproduzir a dinâmica da regulação gênica.

2.9 AGRUPAMENTO

Agrupamento é uma técnica que objetiva descobrir conjuntos de categorias que mantém dados similares em um mesmo grupo (ARABIE; HUBERT; SOETE, 1996).

Técnicas de agrupamento são aplicadas em diversas áreas, tais como detecção de anomalias (MÜNZ; LI; CARLE, 2007), segmentação de imagem médica (NG *et al.*, 2006), segmentação de mercado (DOLNICAR, 2002), entre outras. Além disso, técnicas de agrupamento são amplamente utilizadas em bioinformática, em áreas tais como a seleção de genes (YANG; HUANG; LIU, 2021) e detecção de novos tipos celulares (CHEN; NING; SHI, 2019).

Tais técnicas agrupam instâncias de dados em subconjuntos de forma que instâncias semelhantes sejam agrupadas, enquanto instâncias diferentes pertencem a grupos diferentes. As instâncias são organizadas em representações que caracterizam a população que está sendo amostrada. Formalmente, a estrutura do agrupamento é representada como um conjunto de subconjuntos $C = \{C_1, \cdots C_k\}$ de S, tal que: $S = \bigcup_{i=1}^{k} C_i$ e $C_i \bigcap C_j = \emptyset$ para $i \neq j$. Consequentemente, qualquer instância em S pertence a exatamente um e somente um subconjunto (ROKACH; MAIMON, 2005).

A determinação de similaridade entre instâncias pode ser feita utilizando-se diversas métricas. Existem dois tipos principais de medidas para estimar essa relação: medidas de distância e medidas de similaridade (ROKACH; MAIMON, 2005).

Uma das métricas mais comuns para medidas de distância de atributos numéricos é a métrica de Minkowski. Dadas duas instâncias p-dimensionais $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ e $x_j = (x_{j1}, x_{j2}, \cdots, x_{jp})$, a distância entre duas instâncias de dadas pode ser calculada como (HAN; KAMBR, 2001):

$$d(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g)^{1/g} \tag{2.7}$$

A distância Euclidiana entre dois objetos é alcançada quando $g = 2$. Se $g = 1$, a soma das distâncias absolutas paraxiais é obtida (métrica de Manhatan). Por fim, se $g = \infty$, obtém-se a maior das distâncias paraxiais (métrica Chebychev).

A unidade de medida utilizada pode afetar a análise do agrupamento. A fim de evitar a dependência na escolha das unidades de medidas, os dados devem ser normalizados. A normalização almeja fornecer a todas as variáveis um peso igual. Contudo, se para cada variável é atribuído um peso de acordo com sua importância, então a distância ponderada pode ser computada como:

$$d(x_i, x_j) = (w_1|x_{i1} - x_{j1}|^g + w_2|x_{i2} - x_{j2}|^g + \cdots + w_p|x_{ip} - x_{jp}|^g)^{1/g} \tag{2.8}$$

Diversos algoritmos de agrupamento foram desenvolvidos, cada um com princípio de indução diferente (ROKACH; MAIMON, 2005). FRALEY; RAFTERY (1998) sugerem

dividir os métodos de agrupamento em dois grandes grupos: métodos de partição e métodos hierárquicos. Trabalhos mais recentes costumam incluir outros dois grupos: métodos baseado em densidade e métodos baseados em *grid* (HAN; KAMBER; PEI, 2012). Contudo, neste trabalho utilizamos somente métodos de partição e métodos hierárquicos, tendo em vista sua simplicidade, além dos problemas relacionados ao ajuste de parâmetros e baixo desempenho em alta dimensionalidade apresentado pelos métodos baseados em densidade.

Os métodos de partição realocam instâncias através da troca destas de um *cluster* para outro, começando por uma partição inicial. Tais métodos tipicamente requerem que o número de *clusters* seja definido pelo usuário (ROKACH; MAIMON, 2005).

Os métodos de partição mais frequentemente utilizados são aqueles baseados em minimização de erro. A ideia é encontrar uma estrutura de agrupamento que minimiza um certo critério de erro que quantifica a distância de cada instância, tanto para outra instância quanto para um dado centro, quando aplicável. O critério mais conhecido é a soma de erros quadráticos (SSE - do inglês *Sum of Squared Error*) (ROKACH; MAIMON, 2005), que quantifica a distância Euclidiana quadrática total das instâncias.

O algoritmo mais comum que utiliza o princípio de partição e o SSE é o K-Means. Esse algoritmo particiona os dados em K *clusters* $(C_1, C_2, \cdots, C_K)$ representados por seus centros ou médias. O centro de cada *cluster*, para uma dimensão, é calculado como a média de todas as instâncias pertencente àquele *cluster*, dado por:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q \tag{2.9}$$

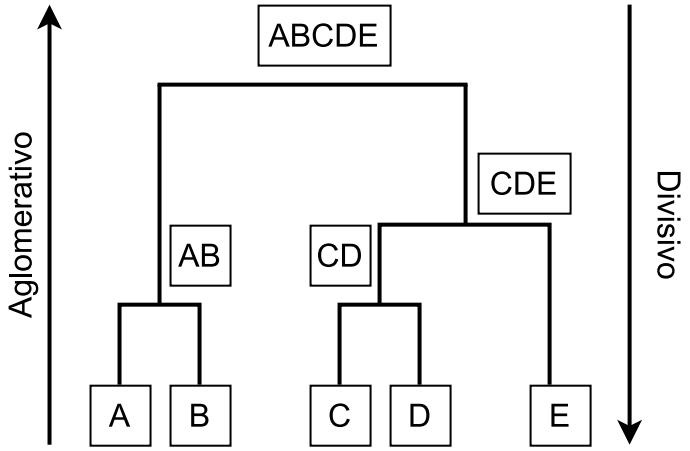
Onde N_k é o número de instâncias pertencente ao *cluster* k, μ_k é a média do *cluster* k e x_q são os registros de dados.

Já os métodos hierárquicos constroem os grupos através da partição recursiva de instâncias, seja de maneira *top-down* ou *bottom-up*. Esses métodos podem ser divididos em (i) agrupamento hierárquico aglomerativo, e (ii) agrupamento hierárquico divisivo. Suas ilustrações, denominadas dendogramas, são apresentadas na Figura 21. Enquanto no primeiro cada objeto inicialmente representa um *cluster* por si só e os demais *clusters* são unificados até que a estrutura do *cluster* desejado seja obtido, no segundo todos os objetos pertencem a um único *cluster* e então são divididos em sub-*clusters*.

Contudo, conforme ressaltado anteriormente, em muitos casos faz-se necessária a determinação explícita do número de *clusters* desejado. Para auxiliar nessa tarefa, diversas medidas podem ser consideradas. Uma das mais comuns é o coeficiente de silhueta.

O coeficiente de silhueta fornece informação sobre a medida do quão bem cada dado está adequado em seu *cluster*, combinando informação sobre a coesão (o quão próximo um dado está dos outros dados em seu próprio *cluster*) e a separação (o quão distante um dado está de outros dados de outros *clusters*)

Figura 21 – Dendogramas dos agrupamentos hierárquicos aglomerativo e divisivo.



Fonte: Elaborado pelo autor (2024).

Assumindo que os dados foram agrupados, para o ponto de dados $i \in C_I$, a distância média entre i e todos os outros pontos de dados no mesmo *cluster* pode ser definida por:

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j) \tag{2.10}$$

onde $|C_I|$ é o número de pontos pertencentes ao *cluster* C_I, e d(i,j) é a distância entre os dados i e j no *cluster* C_I. A dissimilaridade média do ponto i para algum *cluster* C_J pode ser definida como a média da distância de i para todos os pontos em C_J (onde $C_J \neq C_I$). Para cada dado $i \in C_I$, a menor média de distância de i para todos os dados em qualquer outro *cluster* é expresso por:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j) \tag{2.11}$$

O *cluster* com a menor média de dissimilaridade é dito *cluster* vizinho. A silhueta de um ponto de dados i é então, dado por:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, se |C_I| > 1 \tag{2.12}$$

Caso $|C_I| = 1$, então $s(i) = 0$.

Isso pode ser reescrito como:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{se a(i)} < \text{b(i)} \\ 0, & \text{se a(i)} = \text{b(i)} \\ b(i)/a(i) - 1, & \text{se a(i)} > \text{b(i)} \end{cases} \tag{2.13}$$

Com isso, *s(i)* está definido no intervalo [-1, 1]. Por fim, o coeficiente de silhueta é definido como o valor máximo da média de *s(i)* sobre todos os dados do conjunto inteiro, dado por:

$$SC = \max_{k} \tilde{s}(k) \tag{2.14}$$

onde $\tilde{s}(k)$ representa a média *s(i)* sobre todos os dados do conjunto de dados inteiro para um número específico de *clusters k*.

De maneira simplificada, o cálculo do coeficiente de silhueta pode ser resumido em, para cada dado, calcula-se:

1. distância média para todos os outros dados do mesmo *cluster* (coesão), e
2. distância média para todos os outros dados do *cluster* vizinho mais próximo (separação).
3. determina-se o coeficiente de silhueta $\frac{\text{(separação - coesão)}}{max(\text{separação, coesão})}$

O melhor número de *clusters* é, portanto, aquele que maximiza o valor do coeficiente de silhueta.

Outra forma de determinar a qualidade de um agrupamento é através do índice de Davies-Bouldin (DBI - do inglês *Davies-Bouldin Index*). O DBI é uma métrica de validação calculada como a medida de similaridade média de cada *cluster* com o seu *cluster* mais similar. Neste contexto, similaridade é definida como a razão entre as distâncias intercluster e intracluster. Para um conjunto de dados $X = \{X_1, X_2, X_3, \cdots, X_N\}$, o DBI para k número de *clusters* pode ser calculado como:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max \left(\frac{\Delta(X_i) + \Delta(X_j)}{\delta(X_i, X_j)} \right) \tag{2.15}$$

onde ΔX_k é a distância intracluster do cluster X_k e $\delta(X_i, X_j)$ é a distância intercluster entre os *clusters* X_i e X_j. De maneira oposta ao coeficiente de silhueta, quanto menor for o DBI, melhor é o dado número *clusters k*.

Além disso, é possível criar grupos levando em consideração o coeficiente de correlação dos dados. Correlação pode ser definida como o grau de associação entre duas variáveis (COHEN *et al.*, 2009). Existem diversos coeficientes de correlação para lidar com características especiais, tais como tipos de variáveis, e existem outras medidas de associação para variáveis nominais e ordinais. A correlação entre duas variáveis comumente varia entre -1 e +1, onde 0 significa não haver correlação, -1 correlação negativa perfeita e +1 correlação positiva perfeita (AKOGLU, 2018). Além disso, é comum determinar a força da correlação conforme (AKOGLU, 2018):

- $\pm$ 1: perfeita
- $>\pm$]0,7 a 0,9]: correlação muito forte
- $\pm$]0,4 a 0,7]: correlação forte
- $\pm$]0,3 a 0,4]: correlação moderada
- $\pm$]0,2 a 0,3]: correlação fraca
- $\pm$]0,1 a 0,2]: correlação desprezível
- $\pm$ [0 a 0,1]: sem correlação

Um dos coeficientes de correlação mais conhecidos é o de Pearson (COHEN *et al.*, 2009), definido como:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} = \frac{cov(X,Y)}{\sqrt{var(X) \cdot var(Y)}} \tag{2.16}$$

onde $x_1, x_2, \cdots x_n,\ y_1, y_2, \cdots, y_n$ são os valores medidos de ambas as variáveis e $\overline{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ e $\overline{y} = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i$ são as médias aritméticas de ambas as variáveis. O coeficiente de correlação de Pearson fornece uma indicação da força da relação linear entre duas variáveis x e y.

Uma opção para identificação de correlações não lineares é o coeficiente de correlação de Spearman. O coeficiente de correlação de Spearman é uma medida não paramétrica de correlação que avalia com que intensidade a relação de duas variáveis pode ser descrita pelo uso de uma função monótona. Para uma amostra de tamanho n e dados X_i e Y_i, o coeficiente de correlação de Spearman pode ser definido como:

$$r_s = \rho_{rg_X, rg_Y} = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{2.17}$$

onde $cov(rg_X, rg_Y)$ é a covariância das variáveis em postos e σ_{rg_X} e σ_{rg_Y} são os desvios padrão das variáveis em postos.

Além disso, é possível utilizar o coeficiente de correlação τ de Kendall. Este coeficiente também é uma medida de correlação de postos, assim como o Spearman. Considerando um conjunto de observações das variáveis aleatórias conjuntas X e Y, tal que todos os valores de (x_i) e (y_i) sejam únicos, qualquer par de observações (x_i, y_i) e (x_j, y_j), em que $i \neq j$, é dito concordante se as classificações de ambos os elementos concordarem uma com a outra. A concordância ocorre se $x_i > x_j$ e $y_i > y_j$ ou se $x_i < x_j$ e $y_i < y_j$. Os casos de discordância ocorrem se $x_i > x_j$ e $y_i < y_j$ ou se $x_i < x_j$ e $y_i > y_j$. Se $x_i = x_j$ ou $y_i = y_j$ o par não é nem concordante, nem discordante. Desta forma, o coeficiente τ de Kendall é definido como:

$$\tau = \frac{(\text{quantidade de pares concordantes}) - (\text{quantidade de pares discordantes})}{n(n-1)/2} \quad (2.18)$$

Dado isto, é possível agrupar dados de acordo com um *threshold* de coeficiente de correlação. Por exemplo, quaisquer duas expressões gênicas que possuam módulo de correlação maior ou igual a um dado *threshold* podem compor um mesmo grupo. Essa ideia de utilizar o coeficiente de correlação para agrupamento de expressões gênicas é proposta no presente trabalho.

2.10 *ENSEMBLE*

Ensembles envolvem o emprego de múltiplos modelos e combinam suas previsões individuais a fim de obter previsões confiáveis e mais precisas (KUMAR; KUMAR, 2012). DIETTERICH (2000) lista três razões para os benefícios dos *ensembles*: razão estatística, computacional e representacional. A razão estatística emerge quando a quantidade de dados de treinamento disponíveis é pequeno quando comparado ao tamaphno do espaço de hipóteses. Já a razão computacional está associada ao fato de que muitos algoritmos de aprendizado trabalham realizando buscas locais que podem estagnar em ótimos locais. Por fim, a razão representacional diz respeito ao fato de que na maioria das aplicações de aprendizado de máquina, a função verdadeira f não pode ser representada em nenhuma das hipóteses no espaço de hipóteses (DIETTERICH, 2000). Ainda, segundo JAIN; DUIN; MAO (2000), um projetista pode ter acesso a vários classificadores diferentes, cada um desenvolvido em um contexto diferente e para uma representação/descrição diferente do mesmo problema. Exemplos disso estão na identificação de pessoas pela voz, rosto e caligrafia (KUMAR; KUMAR, 2012). Além disso, em muitos casos, mais de um único conjunto de treinamento está disponível. Tais conjuntos podem ser coletados em momentos ou ambientes diferentes. Diferentes modelos treinados nos mesmos dados podem não apenas diferir em seus desempenhos globais, mas também podem apresentar fortes diferenças locais (KUMAR; KUMAR, 2012). Cada modelo pode ter sua própria região no espaço de características onde apresenta o melhor desempenho.

Diferentes estratégias de *ensemble* podem ser aplicadas (CORONA *et al.*, 2009). Uma abordagem envolve o uso de diferentes modelos para tomar uma decisão única sobre o padrão de dados. Nesse caso, modelos treinados com o mesmo conjunto de dados, apresentando desempenho diferente, ajudam a manter a diversidade entre os modelos base.

Em relação aos métodos para a construção de *ensembles*, segundo DIETTERICH (2000), diversos métodos tem sido desenvolvidos. Dentre aqueles de propósito geral, destacam-se o (i) voto Bayesiano, (ii) a manipulação de exemplos de treinamento, (iii) a manipulação das características de entrada, (iv) a manipulação dos alvos de saída, e (v) injeção de aleatoriedade. Para (i), o *ensemble* consiste de todas as hipóteses do espaço de

hipóteses ponderadas por sua probabilidade posterior. Pela regra de Bayes, a probabilidade posterior é proporcional à probabilidade dos dados de treinamento vezes a probabilidade anterior. Em (ii) os *ensembles* são construídos a partir da manupulação dos exemplos de treinamento para a geração de múltiplas hipóteses. O algoritmo de aprendizado é executado diversas vezes, cada uma com um conjunto diferente de amostras de treinamento. Já em (iii), manipula-se o conjunto de características de entrada disponíveis a fim de formar diferentes conjuntos de características de entradas e fornecendo-os para o algoritmo de aprendizado. Para (iv), a construção de *ensembles* de classificadores é realizada a partir da manipulação dos valores de saída fornecidas para o algoritmo de aprendizado. Isso pode ser feito, por exemplo, separando o número de classes K em dois subconjuntos, criando dois novos problemas de aprendizado. Por fim, (v) funciona através da injeção de aleatoriedade no algoritmo de aprendizado. Um exemplo disso pode ser obseervado no algoritmo de *backpropagation* para o treinamento de redes neurais, onde os pesos iniciais da rede são atribuídos de maneira aleatória. Se o algoritmo é aplicado às mesmas amostras de treinamento mas com pesos iniciais diferentes, o classificador resultante pode ser significativamente diferente.

Em resumo, utilizar o conhecimento combinado de vários modelos com diferentes características, os *ensembles* tendem a ser capazes de fornecer resultados melhores e mais robustos. O uso de tal técnica é defendida principalmente no que diz respeito a falta de dados de formação de qualidade para uma avaliação realística e a melhoria do desempenho, em relação à um modelo único (KUMAR; KUMAR, 2012).

2.11 MODELAGEM E INFERÊNCIA DE REDES DE REGULAÇÃO GÊNICA

Diversos modelos computacionais têm sido desenvolvidos para a análise de redes regulatórias. Segundo (KARLEBACH; SHAMIR, 2008), esses modelos podem ser divididos em três classes: modelos lógicos, modelos contínuos e modelos de nível de molécula única.

No que diz respeito ao último tipo, sua introdução seguiu a observação de que a funcionalidade de redes de regulação são comumente afetadas por ruídos. Neste tipo de modelo, almeja-se explicar a relação entre a estocasticidade e a regulação gênica, tendo em vista a interação entre as moléculas individuais (KARLEBACH; SHAMIR, 2008).

Aqui, discute-se majoritariamente as características dos modelos lógicos e contínuos uma vez que além de serem as duas formas mais comuns de modelagem de GRNs, as análises associadas à relação entre estocasticidade e a regulação gênica integram os protocolos mais recentes de modelagem e inferência de GRNs, principalmente quando considerado o perfilamento por scRNA-Seq (SANGUINETTI *et al.*, 2019).

Os modelos lógicos constituem a mais simples modelagem de GRNs e descrevem as redes regulatórias qualitativamente. Eles permitem que os usuários obtenham uma compreensão básica das diferentes funcionalidades de uma determinada rede sob diferentes

condições. Sua natureza qualitativa os torna flexíveis e fáceis de ajustar aos fenômenos biológicos, embora possam responder apenas a questões qualitativas. Foram primeiramente introduzidos por Kauffman e Thomas (GLASS; KAUFFMAN, 1973; KAUFFMAN *et al.*, 1993).

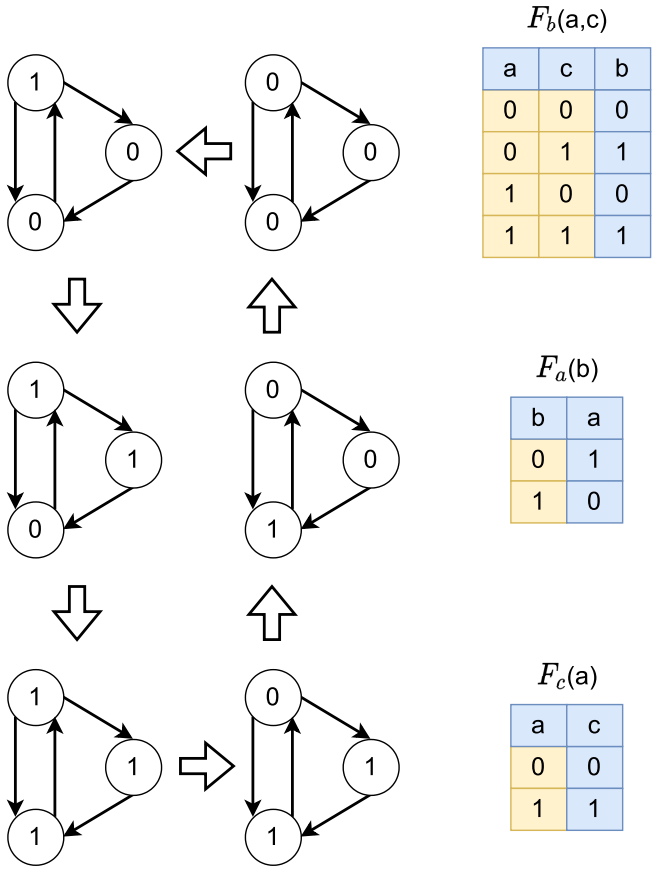
Um exemplo dos *insights* profundos que o exame qualitativo de modelos lógicos de redes regulatórias pode prover é a reconstrução das GRNs que controlam o desenvolvimento de embriões de ouriço-do-mar (DAVIDSON *et al.*, 2002; SMITH; THEODORIS; DAVIDSON, 2007). Neste tipo de modelo, cada entidade do sistema (genes, proteínas e pequenas moléculas), em qualquer ponto do tempo, são representados como um nível discreto e o desenvolvimento temporal do sistema é comumente assumido ser síncrono. A modelagem discreta permite que os pesquisadores confiem em conhecimento qualitativo e os modelos podem ser analisados utilizando uma ampla gama de métodos matemáticos bem estabelecidos (KARLEBACH; SHAMIR, 2008). Dentre os modelos lógicos, focaremos nossa atenção nos Booleanos tendo em vista o determinismo associado, explorado neste trabalho, e a consequente não necessidade de probabilidades associadas como ocorre nos modelos Bayesianos.

Nas redes Booleanas, uma entidade pode ter dois estados, chamados de ativo (1) e inativo (0). O valor das variáveis representa o nível de expressão de cada gene e a cada variável está associada uma função que determina o próximo estado (nível de concentração da proteína no instante seguinte) (MCCALL, 2013).

O vetor binarizado que descreve os níveis de todas entidades é chamado de estado do sistema ou estado global. Assume-se também que a atualização do sistema é feita de maneira síncrona, de tal maneira que a cada passo no tempo, o nível de cada entidade é determinada de acordo com os níveis de seus reguladores no ponto anterior do tempo e de acordo com a função de regulação, como apresentado na Figura 22, onde é possível observar os estados discretos encontrados e suas transições (tripletes de valores discretos) e sua evolução ao longo do tempo (setas entre as tripletes), além das funções de atualização de cada espécie ($F_a(b)$, $F_b(a, c)$ e $F_c(a)$) que determinam o próximo estado discreto de cada espécie.

As redes Booleanas não modelam corretamente a dinâmica do fator de transcrição que desregula a si mesmo, devido ao seu nível de detalhes limitado (KARLEBACH; SHAMIR, 2008). Outro problema é que, como o número de estados globais cresce exponencialmente com o número de entradas, os modelos se tornam computacionalmente caros. Por outro lado, os modelos baseados em redes Booleanas simplificam a estrutura e a dinâmica da regulação gênica. As redes inferidas provêm uma medida qualitativa dos mecanismos regulatórios gênicos (SANGUINETTI *et al.*, 2019) que, apesar de sua simplicidade, podem representar fenômenos significativos. Além disso, é possível obter diversos usos práticos, como a identificação de drogas para tratamento de câncer através

Figura 22 – Representação de um modelo Booleano. À esquerda, os estados em cada passo do tempo, para as espécies a (nó mais acima), b e c, no sentido anti-horário. À direita, as funções de atualização que descrevem as regras do modelo. Segundo esta representação, se a estiver no estado 1 e c estiver no estado 0, no próximo passo o estado de b será 0. Setas finas indicam os reguladores de cada nó. Os intervalos de tempo são representados por setas grossas. O estado global do modelo é a combinação dos três estados da entidade.



Fonte: Adaptado de (KARLEBACH; SHAMIR, 2008).

da inferência dos relacionamentos entre os genes a partir de dados experimentais como os perfis de expressão gênica (MCCALL, 2013).

O método mais difundido para modelar sistemas dinâmicos em ciência e engenharia são as equações diferenciais ordinárias (EDOs), cuja aplicação também é encontrada na inferência das GRNs (JONG, 2002). Este tipo de modelo utiliza as concentrações de RNAs, proteínas e outras moléculas modelando-as através de variáveis dependentes do tempo. Interações regulatórias levam à forma de relações funcionais e diferenciais entre as variáveis de concentração (SANGUINETTI *et al.*, 2019). Mais especificamente, a regulação gênica é modelada por equações que expressam a taxa de produção de um componente do sistema em função das concentrações de outros componentes. Equações de taxa têm a forma:

$$\frac{dx_i}{dt} = f_i(\mathbf{x}), 1 \leq i \leq m \tag{2.19}$$

onde $\mathbf{x} = [x_1(t), ..., x_n(t)] \geq \emptyset$ é o vetor de concentração de proteínas, mRNAs ou pequenas moléculas, e $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ é comumente uma função não linear. A taxa de síntese do

i-ésimo componente é vista dependente das concentrações de $\mathbf{x}$, possivelmente incluindo x_i e pode ser estendido para incluir concentrações u ≤ 0 de componentes de entradas (tais como suprimento de nutrientes externos) (JONG, 2002).

Os atrasos decorrentes do tempo necessário para completar a transcrição, tradução e difusão para o local de ação de uma proteína também podem ser representados:

$$\frac{dx_i}{dt} = f_i(x_1(t-\tau_{i1}), ..., x_n(t-\tau_{in})), 1 \leq i \leq n \tag{2.20}$$

onde $\tau_{i1}, ..., \tau_{in} > 0$ representam os atrasos.

Um modelo de GRNs baseado em sistemas de equações diferenciais ordinárias bastante difundido é o *S-System* (SAVAGEAU, 1998), obtendo sucesso no processo de inferência de GRNs (KIMURA *et al.*, 2005; SPIETH *et al.*, 2004; ALMEIDA; VOIT, 2003; KIKUCHI *et al.*, 2003; NOMAN; IBA, 2005; THOMAS *et al.*, 2004). Formalmente, os *S-Systems* são um tipo específico de sistemas de equações diferenciais que possuem uma forma canônica simples, amplamente utilizada no contexto de extração de modelos de redes de regulação gênica a partir de dados *microarray* na forma de séries temporais (KIMURA *et al.*, 2005; SPIETH *et al.*, 2004; KUTALIK; TUCKER; MOULTON, 2007; ANDO; IBA, 2003), e adequado para descrever aderentemente redes biológicas (VOIT, 2000). O modelo dos *S-Systems* é dado por:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{r_{i,j}} - \beta_i \prod_{j=1}^{N} x_j^{s_{i,j}}, (i = 1, ..., N) \tag{2.21}$$

onde x_i são as variáveis de estado. Além disso, dois parâmetros positivos constantes estão presentes (α_i e β_i). $r_{i,j}$ e $s_{i,j}$ são parâmetros exponenciais chamados de ordem de cinética. Se $r_{i,j} > 0$, o gene j induzirá uma expressão do gene i. Entretanto, se $g_{i,j} < 0$, o gene j inibirá a expressão do gene i. O parâmetro $s_{i,j}$ tem o efeito contrário de $r_{i,j}$. O *S-system* é um modelo quantitativo e possui uma rica estrutura, capaz de capturar dinâmicas em diversos sistemas bioquímicos (WANG; QIAN; DOUGHERTY, 2010).

A modelagem matemática de processos biológicos fornece compreensões profundas sobre os sistemas celulares. Embora modelos quantitativos e contínuos, como equações diferenciais, tenham sido amplamente utilizados, seu uso é difícil em sistemas onde o conhecimento dos parâmetros cinéticos são escassos (SANGUINETTI *et al.*, 2019). Por outro lado, uma riqueza de nível qualitativo molecular dados sobre componentes e interações individuais podem ser obtidos a partir da literatura, auxiliando na criação de abordagens qualitativas como os modelos Booleanos (SANGUINETTI *et al.*, 2019).

Dentre os métodos estado da arte para inferência de GRNs, aqui as explicações são concentradas naqueles apresentados em Pratapa et al. (PRATAPA *et al.*, 2020), que provêm uma série de problemas *benchmark* e um *framework* (BEELINE[5]) de avaliação para

[5] https://murali-group.github.io/Beeline/BEELINE.html

a inferência de redes de regulação gênica utilizando dados oriundos de perfilamento por scRNA-Seq. Detalhes sobre o *framework* serão discutidos em seções de capítulos posteriores. Em PRATAPA *et al.* (2020), 12 algoritmos estado da arte: PIDC, GENIE3, GRNBOOST2, PPCOR, SCODE, GRISLI, SINGE, SCNS, LEAP, SINCERITIES, GRNVBEM e SCRIBE são comparados. Aqui não discute-se sobre o SCRIBE pois não foi possível utilizá-lo dentro do *framework*. Dentre estes algoritmos, GENIE3, GRNBOOST2, PIDC, SINCERITIES e PPCOR são apontados como os melhores segundo a avaliação realizada (PRATAPA *et al.*, 2020).

Gene Network Inference with Ensemble of Trees (GENIE3) (IRRTHUM *et al.*, 2010) é um método baseado em árvores, similar a *random forests* e é usado para predizer a expressão de todos os outros genes. GENIE3 foi desenvolvido para *bulk* RNA-Seq e trata o problema de reconstruir uma GRN de N genes como N problemas de regressão, tentando determinar o subconjunto de genes cujos perfis de expressão são os mais preditivos do perfil de expressão de um gene alvo. Os genes deste subconjunto são classificados com base em pesos que são calculados como a soma da redução da variância total da variável de saída devido à divisão. Portanto, *ranks* maiores significam regulações regulatórias mais fortes. Os experimentos são realizados no conjunto de dados DREAM4 e a conclusão é que GENIE3 pode prever a direção das relações até certo ponto, embora apenas explore medições de estado estacionário.

GRNBOOST2 (MOERMAN *et al.*, 2019) é similar ao GENIE3 porém mais rápido e adequado especialmente para conjuntos de dados maiores. Esse método usa *stochastic gradient boosting machine regression* com regularização de parada antecipada para selecionar os reguladores mais importantes para cada gene no conjunto de dados e inferir a rede. A qualidade das GRNs inferidas é avaliada utilizando os problemas do DREAM5. A aceleração do GRNBOOST2 no conjunto de dados scRNA-seq é alcançada por dois fatores. Em primeiro lugar, o efeito de redução de viés do aumento de gradiente permite o uso de árvores de decisão mais rasas do que a floresta aleatória. Em segundo lugar, usando a parada antecipada, GRNBOOST2 construiu mais de 80% menos árvores de decisão no total do que GENIE3. O *framework* Arboreto, que fornece tanto o GENIE3 quanto o GRNBOOST2, dimensiona ambos os algoritmos de inferência GRN de forma aproximadamente linear em relação aos recursos computacionais.

Gene Regulation Inference for Single-cell with Linear differential equations and velocity Inference (GRISLI) (AUBIN-FRANKOWSKI; VERT, 2020) toma o tempo experimental ou *pseudotime* estimado das células como entrada e estima como o valor de expressão de cada gene muda à medida em que cada célula passa por um processo dinâmico, chamado de velocidade. Em seguida, a estrutura do GRN subjacente é calculada resolvendo um problema de regressão esparsa que relaciona a expressão gênica e os perfis de velocidade de cada célula usando um formalismo linear baseado em EDOs. O GRISLI é aplicado em vários conjuntos de dados de scRNA-Seq, como a reprogramação

de fibroblastos embrionários murinos para miócitos, com 373 células, a diferenciação das células ES humanas para células endodérmicas definitivas, com 758 células e um conjunto de 3.696 células pancreáticas murinas para estudar endocrinogene pancreático. O GRISLI é comparado ao SCODE, TIGRESS e GENIE3 e os autores mostram que, em dados reais, o GRISLI supera o SCODE e o TIGRESS em dados de scRNA-seq de humanos e murinos.

Uma rede Bayesiana codifica dependências condicionais entre variáveis aleatórias que são nós da rede. Cada nó é uma tabela de probabilidade condicional (dados discretos) ou um modelo de regressão (variáveis contínuas), que especifica a probabilidade de obter um determinado resultado para o nó, dados os valores de seus nós pais (CHEN; MAR, 2018). *GRN with variational Bayesian Expectation-Maximization* (GRNVBEM) (SANCHEZ-CASTILLO *et al.*, 2018) infere uma rede Bayesiana usando um modelo autorregressivo de primeira ordem. Este modelo estima a mudança de dobra de um gene em um momento específico como uma combinação linear da expressão dos reguladores do gene na rede Bayesiana no momento anterior. Os resultados são avaliados com dados de qPCR de célula única e RNA-Seq para células embrionárias de camundongos e células hematopoiéticas em dados de peixe-zebra. Os resultados mostram que o método GRNVVBEM infere com sucesso o papel ativo de Tcfap2c e Sall4 durante a transição ICM para endoderme primitivo. Para dados de RNA-Seq de célula única, foi inferido um módulo GRN que inclui potenciais genes-chave na biogênese mitocondrial e diferenciação de granulócitos. Além disso, o GRNVBEM apresentou um desempenho robusto ao considerar os efeitos do *pseudotime* em termos de especificidade.

Lag-based Expression Association for Pseudotime-series (LEAP) (SPECHT; LI, 2017) usa dados ordenados pseudotemporalmente e calcula a correlação de Pearson de contagens de leitura mapeada normalizada em janelas temporais de tamanho fixo com diferentes atrasos. A pontuação registrada para um par de genes é a máxima correlação de Pearson sobre todos os valores de atraso que o método considera. O método fornece como saída uma rede direcionada (PRATAPA *et al.*, 2020). Para verificar a capacidade do LEAP em detectar relações regulatórias biologicamente verdadeiras, uma rede *Mus musculus* foi considerada. Para comparação de desempenho, foi calculada uma rede regular baseada em correlação de Pearson sem considerar atrasos de tempo. O LEAP foi capaz de capturar associações que estavam ocultas pelos desfasamentos de tempo. As associações assimétricas detectadas pelo LEAP provavelmente refletem relações regulatórias, pois descrevem qual gene segue outro gene na expressão.

Partial information decomposition (PID) foi introduzida para medir dependências estatísticas em uma tripla de variáveis simultaneamente. PID trabalha particionando a informação provida de duas fontes de variáveis sobre uma variável alvo como três categorias: redundante, única e sinergética (WILLIAMS; BEER, 2010; CHEN; MAR, 2018). Dada uma variável aleatória S e um vetor aleatório $R = \{R_1, R_2\}$, é possível calcular a informação total dada pelo vetor para S, que é dado pela informação mútua $I(S; R_1, R_2)$. R_1 pode

prover informação que R_2 não fornece, ou vice-versa (informação única). Além disso, R_1 e R_2 podem prover a mesma informação (redundância). Finalmente, a combinação de R_1 e R_2 podem prover informações que não seriam obtidas se analisadas separadamente (sinergia). PIDC (CHAN; STUMPF; BABTIE, 2017) usa PID entre todos os pares de genes (x, y) a fim de obter um componente único e redundante (z). Então, computa a razão entre o componente único e a informação mútua. A soma dessa razão sobre todos os demais genes z é a contribuição única proporcional entre x e y. O método então usa limiares por gene para identificar as interações mais importantes para cada gene. Os autores consideram cinco conjuntos de dados *in silico* de 50 genes e cinco de 100 genes. O PIDC tem um desempenho favorável em comparação com ARACNE, MRNET, MI, PUC, PIDC e CLR, particularmente nas redes maiores. Além disso, qPCR de célula única para estudar progenitores de megacariócitos-eritróides durante a hematopoiese humana e conjuntos de dados de desenvolvimento hematopoiético embrionário foram usados. O algoritmo foi capaz de capturar as interações mais importantes para cada nó, em vez das maiores dependências em todo o conjunto de dados.

O método *partial and semi-partial correlation* (PPCOR) (KIM, 2015) computa os coeficientes de correlações parciais e semi-parciais para todo par de genes em relação a todos os outros genes. Além disso, calcula-se também o p-valor de cada correlação. O princípio subjacente de redes de correlação é que, se dois genes estão coexpressados, então assume-se que eles participam mutuamente em uma interação regulatória (CHEN; MAR, 2018).

O *Single Cell Network Synthesis toolkit* (SCNS) é uma ferramenta de propósito geral para reconstrução e análise de modelos a partir de dados de expressão gênica perfilados por scRNA-Seq (WOODHOUSE *et al.*, 2018). Esse método utiliza dados scRNA-Seq sobre um curso no tempo e determina as expressões booleanas que conduzem a progressão e transformação de estados de células iniciais para estados de células posteriores com um esquema de atualização assíncrono. Para avaliar a eficiência do SCNS, quatro conjuntos de dados foram considerados: dois conjuntos de dados sintéticos de tamanhos variados, o conjunto de dados de pré-implantação (WOODHOUSE *et al.*, 2018), e o conjunto de dados de sangue embrionário (MOIGNARD *et al.*, 2015), com parâmetros diferentes. Vários modelos de previsão foram validados experimentalmente, demonstrando que os genes HoxB4 e Sox17 regulam diretamente o fator hematopoiético Erg, e que forçar a expressão de Sox7 bloqueia o desenvolvimento do sangue.

SCODE (MATSUMOTO *et al.*, 2017) reconstrói GRNs com dinâmica regulatória baseado na transformação de EDOs lineares a partir de EDOs lineares de parâmetro fixo. A expressão relacional pode ser estimada analiticamente e de maneira eficiente através de regressão linear. Utilizando-se de redução dimensional, a dinâmica da expressão pode ser reconstruída com um pequeno número de padrões, que leva a uma complexidade de tempo reduzida do algoritmo. SCODE foi aplicado a três conjuntos de dados de scRNA-Seq

durante a diferenciação. Os resultados mostram que SCODE pode otimizar EDOs de maneira bem sucedida, de tal maneira que as EDOs representem as dinâmicas de expressão observadas. Na validação da rede inferida, os valores de área sob a curva (AUC - do inglês *area under the curve* foram superiores àqueles de outros métodos em praticamente todos os casos. Além disso, SCODE é mais rápido que o GENIE3, que não utiliza informação temporal.

SINCERITIES (GAO *et al.*, 2018) foi desenvolvido para lidar com dados scRNA-Seq e computa as mudanças temporais na expressão de cada gene a partir da distância das distribuições marginais entre dois pontos de tempo consecutivos usando a estatística de Kolmogorov-Smirnov. As relações regulatórias são inferidas usando as mudanças na expressão gênica dos fatores de transcrição na janela de um tempo e predizer como as distribuições de expressão dos genes alvo mudam na próxima janela de tempo (casualidade de Granger). Os sinais são inferidos utilizando análises de correlação parcial (PRATAPA *et al.*, 2020). SINCERITIES supera os problemas de grande exigência de conjunto de dados, alta complexidade computacional e ordenação de células difícil, pois a inferência de rede envolve regressão linear regularizada numericamente eficiente e usa dados de seção transversal com carimbo de data/hora diretamente. Quando os intervalos de tempo são curtos, a formulação SINCERITIES pode perder as regulações gênicas devido a respostas gênicas atrasadas. No entanto, como as janelas de tempo na análise de célula única geralmente diferem em horas, esse problema pode não ser proeminente (GAO *et al.*, 2018).

Single-Cell Inference of Networks using Granger Ensembles (SINGE) (DESHPANDE *et al.*, 2019) usa regressão de causalidade Granger para aliviar irregularidades em valores *pseudotime*, pois a distribuição de células em todo o processo dinâmico subjacente pode não ser uniforme. Então, para cada conjunto de parâmetros de entrada, o SINGE realiza regressões e agrega as previsões resultantes usando um método Borda modificado. O SINGE é comparado ao SCODE e SINCERITIES usando dados de scRNA-Seq sintéticos e reais. O SINGE apresentou melhor *precision-recall* e superou os outros métodos. Além disso, o SINGE é mais resiliente ao *dropout* nos dados scRNA-Seq e foi o melhor método GRN ao inferir redes a partir de conjuntos de dados simulados com ruído técnico adicional.

Contudo, além do algoritmo SCNS nenhum dos demais considerados estado da arte por PRATAPA *et al.* (2020), fornecem modelos Booleanos. Por esse motivo, a pesquisa na literatura estendeu-se a buscar aqueles que modelam as GRNs de forma Booleana e levantar motivos pelos quais estes não são considerados estado da arte. Dentre as pesquisas, o trabalho de PUŠNIK *et al.* (2022) apresenta uma revisão e avaliação de abordagens Booleanas para a inferência de GRNs. Neste trabalho, são considerados 5 algoritmos, a saber: REVEAL, Best-Fit Extension, MIBNI, GABNI e ATEN e são realizadas comparações da qualidade das soluções obtidas por estes algoritmos em conjuntos de dados sintéticos obtidos a partir de uma rede de referência de *E. coli*, através do *GeneNetWeaver*, com 16, 32 e 64 genes.

O REVEAL (*REVerse Engineering ALgorithm*) (LIANG; FUHRMAN; SOMOGYI, 1998) é uma abordagem baseada em teoria da informação para a inferência de arquiteturas de redes genéticas. Para cada gene alvo, o algoritmo investiga exaustivamente todas as $\binom{N}{k}$ combinações de conjuntos de relações regulatórias, onde k é o número de reguladores e N é o número de nós. O conjunto de relações regulatórias $\mathscr{R}$ determina completamente o gene alvo x se $\mathscr{R}$ contabiliza toda a entropia de x, ou seja, se a informação mútua entre $\mathscr{R}$ e x é a mesma entropia de x.

O método *Best-Fit Extension* (LÄHDESMÄKI; SHMULEVICH; YLI-HARJA, 2003) é um algoritmo de inferência Booleano baseado em encontrar funções Booleanas que o menor tamanho de erro acerca de funções Booleanas parcialmente definidas através da solução de um problema de inconsistência. Uma função Booleana parcialmente definida (pdBf) é um par de conjuntos T e F contendo os vetores de todos os exemplos verdadeiros e falsos, respectivamente. A função f é consistente se T é um subconjunto de todos os exemplos onde f é verdadeiro ($T \subseteq T(f)$), e F é um subconjunto de todos os exemplos onde f é falso ($F \subseteq F(f)$). O problema de consistência pode ser resolvido somente se a interseção dos conjuntos T e F é um conjunto vazio. Dessa forma, para um dado gene, *Best-Fit Extension* retorna todas ass funções Booleanas com menor tamanho de erro, definido por:

$$\varepsilon(F) = \omega(T \cap F(f)) + \omega(F \cap T(f)) \tag{2.22}$$

onde $\omega(S)$ é a soma dos pesos individuais do vetor no conjunto de todos os exemplos de treino. Estes pesos, por sua vez, podem ser uniformes ou podem depender dos dados de treinamento a fim de lidar melhor com conjuntos de dados desbalanceados.

Com o objetivo de reduzir o tempo computacional e aumentar a acurácia de inferência, MIBNI (*Mutual Information-based Boolean Network Inference* (BARMAN; KWON, 2017) foi introduzido. Inicialmente, MIBNI identifica um conjunto de genes regulatórios iniciais que podem melhor caracterizar a variável alvo. O método identifica um subconjunto ótimo $\mathscr{R}$ com seleção de características baseadas em uma medida multivariada de informação mútua aproximada (BATTITI, 1994). O problema associado é que o número de todos os possíveis conjuntos cresce exponencialmente. Dessa forma, a técnica de seleção de características trabalha incrementalmente sob a premissa de expressões gênicas independentes. A cada iteração, uma nova variável v é adicionada ao conjunto candidato ótimo $\mathscr{R}$ baseado no seguinte critério:

$$arg\max_{v \notin R} M(x, v) - \sum_{r \in R} M(r, v) \tag{2.23}$$

onde x representa o gene alvo. Contudo, é importante destacar que MIBNI limita o tamanho máximo do conjunto $\mathscr{R}$ a 10.

GABNI (*Genetic Algorithm-based Boolean Network Inference* (BARMAN; KWON, 2018) surge para suprir a limitação do MIBNI, onde apenas funções de disjunção e conjunção são consideradas como funções Booleanas. GABNI age quando MIBNI falha em encontrar uma solução ótima para um gene alvo. Dessa forma, GABNI utiliza um algoritmo genético ara selecionar um conjunto ótimo de genes regulatórios. Cada cromossomo é fcomposto de um vetor binário de tamanho N, onde cada elemento define a presença ou ausência do i-ésimo gene como reuglador do gene observado. A cada geração do GA, o fenótipo de cada cromossomo é definido com base na expressão gênica $x(t+1$ e seus potenciais reguladores $r(t) = r_1(t)r_2(t)\cdots r_k(t)$ para todas as *strings* de *bit* $b = b_1b_2\cdots b_k$.

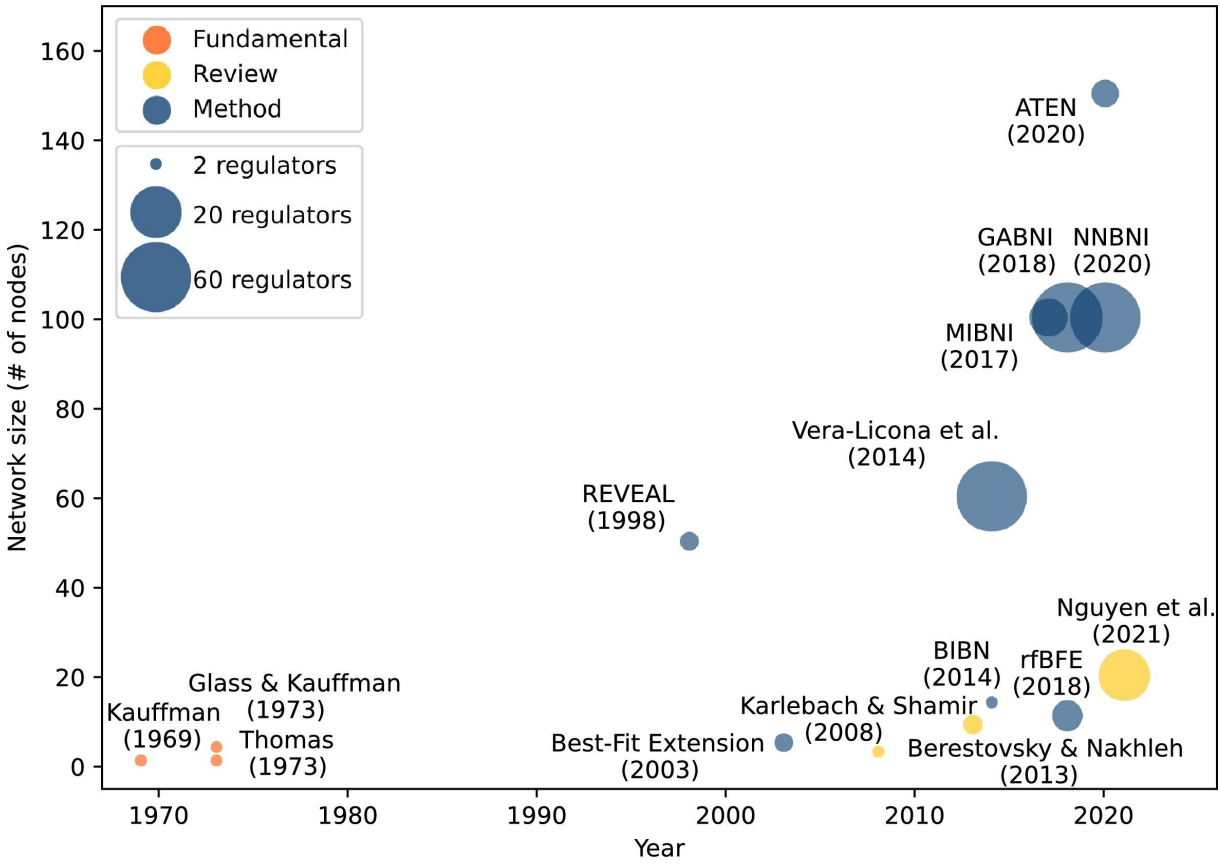
Por fim, ATEN (*AND/OR Tree ENsemble algorithm* (SHI *et al.*, 2020), ao invés de selecionar um subconjunto de genes que podem caracterizar um gene alvo, ATEN foca em produzir uma função Booleana acurada em forma normal disjuntiva empregando um algoritmo *AND/OR Tree Ensemble*. A função Booleana é representada por uma árvore AND/OR em três níveis: (i) a disjunção lógica (OU), (ii) a conjunção (E), e (iii) as folhas da árvore, que podem conter as variáveis Booleanas ou suas negações. ATEN extrai amostras dos dados de séries temporais e, para cada amostra, uma função Booleana para um gene alvo é inferido utilizando *Simulated Annealing* (BERTSIMAS; TSITSIKLIS, 1993). A importância é calculada com base em como a remoção ou adição de um implicante principal à árvore reduz ou aumenta a classificação incorreta.

É importante direcionar a atenção para os métodos GABNI e ATEN, uma vez que utilizam a modelagem de redes Booleanas e utilizam, também, técnicas de computação evolucionista. Contudo, PUŠNIK *et al.* (2022) ressaltam algumas características importantes dentre os métodos comparados e métodos clássicos para a inferência de modelos Booleanos de GRNs, principalmente no que diz respeito ao fato de que modelos Booleanos podem fornecer poder preditivo suficiente em diferentes aplicações, indo em direção oposta ao apresentado em KARLEBACH; SHAMIR (2008), colocando modelos Booleanos no ponto mais baixo na escala de expressividade.

A Figura 23 ilustra a relação entre os algoritmos de inferência de modelos Booleanos, a capacidade de lidar com quantidades crescentes de genes e as limitações associadas ao número máximo de reguladores por gene.

É possível perceber que, desde a introdução dos modelos Booleanos por Kauffman em 1969, com o passar dos anos os métodos de inferência de GRNs Booleanas foram ampliando sua capacidade em lidar com maiores quantidades de genes. Esse problema é comum na área de GRNs Booleanas tendo em vista a complexidade computacional associada ao crescimento exponencial em relação ao número de genes envolvidos. Contudo, mesmo após 2010, muitos algoritmos permanecem na faixa de no máximo 40 genes. Além disso, o gráfico apresentado na Figura 23 também mostra o número máximo de reguladores que cada método consegue lidar. Quando analisados os métodos mais novos (GABNI,

Figura 23 – Evolução da capacidade dos algoritmos lidarem com quantidades de genes ao longo dos anos e as limitações associadas ao número de reguladores máximos.



Fonte: PUŠNIK *et al.* (2022).

NNBNI e ATEN), verifica-se que o limite encontra-se em 150 genes e 60 reguladores. Isso torna-se um grande problema, não só do limite do tamanho da rede, mas também em assumir que um gene possui no máximo 60 reguladores, tendo em vista a quantidade de genes dos organismos, como por exemplo a *E. coli*, com estimados 4.400 genes e o ser humano, com aproximadamente 30.000 genes. Essas características podem justificar o fato de tais algoritmos não estarem incluídos no conjunto de algoritmos apresentados por PRATAPA *et al.* (2020).

2.11.1 Discretização de Dados de Expressão Gênica

A discretização de dados é uma técnica usada em ciência da computação e estatística, frequentemente aplicada como etapa de pré-processamento na análise de dados biológicos. Em geral, o objetivo da discretização de dados de expressão gênica é permitir a aplicação de algoritmos para a inferência de conhecimento biológico que requer dados discretos como entrada (GALLO *et al.*, 2015).

O processo de discretização transforma dados quantitativos em dados qualitativos, isto é, concentrações de mRNA em um número finito de intervalos, obtendo, como resultado,

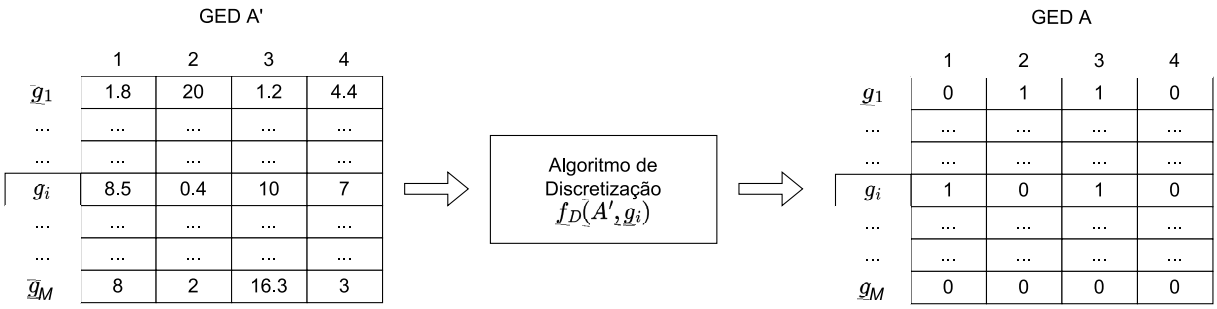
uma partição não sobreposta do domínio contínuo (GALLO *et al.*, 2015). Uma associação entre cada intervalo com um valor discreto é então estabelecida. Na prática, a discretização é uma técnica de redução de dados pois existe um mapeamento de um grande espectro de valores numéricos de expressão gênica para um subconjunto reduzido de valores discretos. A natureza qualitativa dos dados discretos implica que diferentes estratégias de discretização podem resultar em modelos distintos de estado discreto. Portanto, a semântica biológica e a interpretação dos modelos resultantes podem diferir, mesmo quando os dados de valor real subjacentes são os mesmos (MADEIRA; OLIVEIRA, 2005).

A inferência de conhecimento a partir de dados discretos tem várias vantagens quando a análise orientada por dados é realizada. O processo de aprendizagem a partir de dados discretos é mais eficiente e eficaz (RICHELDI; ROSSOTTO, 1995; CHLEBUS; NGUYEN, 1998; CIOS; PEDRYCZ; SWINIARSKI, 1998), exigindo uma quantidade reduzida de dados em comparação com outros métodos que usam valores contínuos (KARLEBACH; SHAMIR, 2008). Além disso, a redução e simplificação dos dados tornam o processo de aprendizado mais rápido, produzindo resultados mais compactos (GARCIA *et al.*, 2012), e permitindo a inferência de modelos de grande porte com maior velocidade de análise (KARLEBACH; SHAMIR, 2008). Além disso, valores discretos são mais fáceis de entender, usar e explicar (GARCIA *et al.*, 2012; KARLEBACH; SHAMIR, 2008). Outra vantagem consiste no fato de que a discretização gera homogeneização de diferentes conjuntos de dados em termos de interpretabilidade. Outra vantagem importante em relação ao uso de dados discretizados é a diminuição do ruído biológico e técnico dos dados brutos (GALLO; CARBALLIDO; PONZONI, 2011), como demonstrado em DIMITROVA *et al.* (2010).

No entanto, a escolha de um método de discretização adequado não é uma tarefa trivial. Em geral, qualquer processo de discretização implica em perda de informação (GARCIA *et al.*, 2012). Por este motivo, a escolha da abordagem de discretização deve considerar não só a natureza intrínseca dos dados biológicos mas também a tecnologia envolvida nas medições e as características particulares do método computacional que será aplicado para a inferência (GALLO *et al.*, 2015).

Para formalizar a definição do método de discretização em questão, considerando uma matriz de dados de expressão gênica A' de M linhas por N colunas, onde a'_{ij} representa o nível de expressão do gene g_i sob a condição j. A matriz A' é definida por seu conjunto de linhas, I, e seu conjunto de colunas J. Uma matriz discretizada A resulta da aplicação de uma função de discretização f_D em A', que mapeia cada elemento a'_{ij} em A' para um dos elementos de um alfabeto Σ. O alfabeto Σ, por sua vez, consiste em um conjunto de k símbolos que podem representar um nível de ativação de gene distinto. Esse processo é apresentado na Figura 24. Segundo GALLO *et al.* (2015), as principais características da discretização de dados de expressão gênica são (i) natureza do problema de aprendizado, (ii) nível de discretização, (iii) tipo de tecnologia de dados, (iv) tipo de amostra, e (v) escopo

Figura 24 – Discretização de dados de expressão gênica. Dada uma matriz de M genes por N condições experimentais A', a aplicação de uma função de discretização $f_D(A', g_i$ retorna uma matriz A discretizada, com $\Sigma = \{0, 1\}$.



Fonte: Adaptado de GALLO *et al.* (2015).

Figura 25 – Principais características da discretização de dados de expressão gênica.

Discretização de dados de expressão gênica				
Supervisão	Supervisionado	Não Supervisionado		
Tipo de Amostra	*Steady State*	Série Temporal		
Tecnologia de Dados	*Microarray*	RNA-Seq	scRNA-Seq	
Nível de Discretização	Binário	Ternário	Multi-Nível	
Escopo de Dados	Pontos	Linha	Coluna	Matriz

Fonte: Adaptado de GALLO *et al.* (2015).

de dados. Esses pontos são discutidos a seguir e resumidos na Figura 25.

Em relação à natureza do problema de aprendizado, consideram-se as abordagens supervisionadas e não supervisionadas. Isso define se a abordagem depende de informações de rótulo de classe para realizar a discretização. Sendo assim, em métodos supervisionados, os valores de g_i são atribuídos em relação às informações de rótulo de classe de um domínio do conhecimento. Entretanto, segundo GALLO *et al.* (2015), a maioria das abordagens de discretização propostas na literatura para dados de expressão gênica são não supervisionados.

O nível de discretização é outro fator importante a ser considerado. No caso mais simples, um alfabeto Σ contém apenas dois símbolos, representando ativação ou inibição. Deste modo, a matriz de expressão é comumente transformada em uma matriz binária,

onde 1 significa ativação e 0, inibição, como apresentado na Figura 24. Outro esquema comumente utilizado considera um conjunto ternário de símbolos {-1, 1, 0}, significando regulação negativa, regulação positiva e sem modificação, respectivamente. Mesmo assim, os valores da matriz A' podem ser discretizadas de maneira multi-nível para um número arbitrário de símbolos.

O nível de discretização depende majoritariamente do algoritmo e inferência que utilizará esse método de discretização. Como discutido anteriormente, toda discretização leva à perda de informação. Em teoria, quanto menor o número de estados discretos possíveis, maior a perda de informação. Em contra partida, muitos estados discretos aproximam-se de um modelo contínuo, o que resulta em perda dos benefícios e vantagens introduzidas pela discretização. Além disso, um aumento na quantidade de símbolos reduz a perda de informação mas aumenta a complexidade computacional do algoritmo (KARLEBACH; SHAMIR, 2008).

O tipo de tecnologia de dados também influencia na seleção de um método de discretização. Em geral, os métodos de discretização foram desenvolvidos para dados de expressão gênica obtidos através de *microarrays* (MADEIRA; OLIVEIRA, 2005; LI *et al.*, 2010; MAHANTA *et al.*, 2012), sem levar em consideração as características particulares dos dados que estão sendo discretizados. Isso permite a aplicação desses métodos tanto para *microarrays*, RNA-Seq e scRNA-Seq, por exemplo. Entretanto, as tecnologias RNA-Seq e scRNA-Seq oferecem diversas vantagens em relação aos *microarrays* convencionais, tais como o baixo sinal de fundo e um aumento na variação das medidas (MARIONI *et al.*, 2008; WANG; GERSTEIN; SNYDER, 2009). Portanto, considerar as características particulares da tecnologia envolvida na extração dos dados biológicos pode levar a um desenvolvimento mais confiável de métodos de discretização (QU *et al.*, 2013) apesar de isso limitar a aplicação em outras plataformas.

Relacionado ao tipo de tecnologia, o tipo de experimento utilizado para obtenção do dado (significado das colunas na matriz de dados) também deve ser considerado. Existem dois tipos de amostragens, uma que considera o estado de equilíbrio (*steady state*), onde os níveis de expressão correspondem a uma situação estática, e os níveis de expressão na forma de séries temporais, que são obtidos durante a fase fenotípica, tais como o ciclo celular (SPELLMAN *et al.*, 1998). Geralmente, nos dados de expressão estáticos, diferentes condições experimentais referem-se a tecidos diferentes, temperaturas, compostos químicos e qualquer outra condição que possa gerar um comportamento regulatório diferente entre os genes amostrados. Por outro lado, nos dados na forma de séries temporais, as linhas da matriz de expressão gênica representam os genes enquanto as colunas representam os pontos no tempo. Por fim, o escopo de dados também deve ser levado em consideração. Os dados de expressão gênica podem ser discretizados segundo o perfil do gene (linhas da matriz), a condição experimental (colunas da matriz) ou a matriz completa. Para séries temporais, comumente utilizam-se as linhas da matriz, tendo em vista que ela representa

a expressão de um mesmo gene ao longo do tempo.

Neste trabalho não utilizaremos métodos de discretização supervisionados, uma vez que as informações de rótulo de classe não estão disponíveis. Desta forma, nossas explicações concentrar-se-ão nos métodos não supervisionados, tanto para *steady-state* quanto para séries temporais, tendo em vista que os métodos de *steady-state* podem ser aplicados em séries temporais. GALLO *et al.* (2015) divide os métodos de discretização para *steady-state* em três grandes grupos, denominados métodos de discretização baseados em (i) métricas, (ii) *ranking* e, (iii) *clustering*.

As abordagens baseadas em métricas usam alguma medida para computar os pontos de corte P para o gene g_i na matriz de dados de expressão gênica A', para determinar o estado discreto correspondente. Em geral, a métrica pode ser computada utilizando diferentes escopos de dados. A abordagem de discretização utilizando métricas é dada por

$$a_{ij} = \begin{cases} 1, & \text{quando } a'_{ij} \geq \delta, \text{ e} \\ 0, & \text{caso contrário.} \end{cases} \tag{2.24}$$

para um alfabeto Σ de dois símbolos, onde δ representa a métrica usada para computação.

O caso mais simples é definir δ como a média da expressão (do escopo de dados escolhido) (MADEIRA; OLIVEIRA, 2005; ZOMAYA; ELLOUMI, 2013). A aplicação deste tipo de discretização foi bem sucedida em (SOINOV; KRESTYANINOVA; BRAZMA, 2003; LI *et al.*, 2006; PONZONI *et al.*, 2007). Outras variações dessa abordagem consistem em utilizar a mediana (*Mid-Range*) ou até mesmo alguma proporção fixa x a respeito do valor máximo do escopo de dados considerado (*X%Max*).

Quando considerado um nível de discretização igual a 3, uma pequena modificação na Equação 2.24 já é suficiente, como

$$a_{ij} = \begin{cases} -1, & \text{se } a'_{ij} < \delta \\ 1, & \text{se } a'_{ij} > \delta, \text{ e} \\ 0, & \text{caso contrário.} \end{cases} \tag{2.25}$$

onde -1 significa regulação negativa, 1 regulação positiva e 0 sem mudanças. Para este caso, δ é um *threshold* para determinar os pontos de discretização (GALLO *et al.*, 2015).

Outra possibilidade é permitir uma discretização multi nível. Isso pode ser alcançado utilizando a discretização de mesma largura (EWD - *Equal Width Discretization*), na qual cada ponto de corte p_r é calculado como $p_{r+1} = p_r + (H - U)/k$, com $p_0 = \text{U}$, onde H e U são o maior e o menor valor do escopo de dados utilizado, respectivamente, e k é o número de intervalos desejados. Desta forma, basta atribuir o símbolo $r \in \Sigma$ correspondente. Exemplos de aplicação desta discretização são encontrados em (MADEIRA; OLIVEIRA, 2005; MAHANTA *et al.*, 2012)

Já as abordagens baseadas em *ranking* assumem inicialmente que os valores de expressão gênica estão ordenados de forma decrescente em uma lista L (GALLO *et al.*, 2015).

A forma mais simples de realizar uma discretização neste caso é atribuir 1 aos primeiros x% valores de L e 0 aos demais. Essa abordagem é conhecida como Top %X (MADEIRA; OLIVEIRA, 2005; GARCIA *et al.*, 2012). Para usar uma discretização multi nível, outra abordagem relacionada é o princípio da frequência igual. Esse método conhecido como discretização de igual frequência (EFD - *Equal Frequency Discretization*) (DOUGHERTY; KOHAVI; SAHAMI, 1995) considera que a lista L é dividida em k segmentos de mesmo tamanho (L/k), contendo a mesma quantidade de dados por segmento. Desta forma, os estados discretos são atribuídos de acordo com a ordem decrescente dos segmentos. Essa abordagem é utilizada em (MADEIRA; OLIVEIRA, 2005; MAHANTA *et al.*, 2012; LONARDI; SZPANKOWSKI; YANG, 2004).

Por fim, o último grupo consiste dos métodos baseados em *clustering* (GALLO; CARBALLIDO; PONZONI, 2011; DIMITROVA *et al.*, 2010; ZOMAYA; ELLOUMI, 2013; LI *et al.*, 2010; MAHANTA *et al.*, 2012). A maneira pela qual isso é alcançado é considerar cada valor a'_{ij} de A' como um elemento de um espaço unidimensional X. Então, o algoritmo de agrupamento é aplicado para os S elementos de X que correspondem a um escopo de dados específico para obter grupos de valores, onde valores pertencendo ao mesmo grupo são atribuídos ao mesmo estado discreto. Os grupos são calculados através da maximização da similaridade entre os elementos de cada *cluster*, enquanto minimizando esse valor entre os elementos em *clusters* diferentes. Uma métrica de qualidade comum para os clusters é o WCSS (*Within-Cluster Sum of Squares*), definido para um esquema de discretização D como

$$\mathrm{WCSS}(D) = \sum_{a'_{ij} \in [p_0, p_1]} |a'_{ij} - \mu_0|^2 + \sum_{r=1}^{k-1} \sum_{a'_{ij} \in (p_r, p_{r+1}]} |a'_{ij} - \mu_r|^2, \tag{2.26}$$

onde μ_r é a média de $a'_{ij} \in (p_r, p_{r-1})$. Menores valores significam maiores similaridades entre os elementos dos *clusters*. Um algoritmo amplamente utilizado para essa tarefa é o *k-means* (MACQUEEN *et al.*, 1967). O *k-means* utiliza distância Euclidiana como medida de similaridade e tenta produzir uma partição de elementos com o WCSS. Os principais pontos desse algoritmo podem ser resumidos como segue: (i) o algoritmo toma um conjunto de pontos S e um inteiro fixo k como entrada, (ii) divide S em k subconjuntos escolhendo um conjunto de k pontos de centróides iniciais, onde os elementos de S são agrupados considerando o centróide mais próximo de seus *clusters*, e (iii) recalcula os centróides a partir dos elementos dentro dos *clusters*. Os passos (ii) e (iii) são iterados até que algum critério de parada seja atingido.

A abordagem mais comum para discretização baseada em *clustering* é utilizar o *k-means* para discretizar tanto os perfis de expressão gênica ou os perfis de condição experimental (LI *et al.*, 2010; MAHANTA *et al.*, 2012), tal como o *Bidirectional K-means* (Bikmeans) (LI *et al.*, 2010), que realiza o *clustering* tanto dos perfis de expressão quanto dos perfis da coluna (informação de tempo ou condição experimental), utilizando o algoritmo *kmeans* (KRISHNA; MURTY, 1999). Isto é, para um dado nível de discretização

Tabela 2 – Discretização por Bikmeans.

		K-means			
		1	2	3	4
Cokmeans	1	$1\times1=1\rightarrow a_{ij}=1$	$a_{ij}=1$	$a_{ij}=1$	$a_{ij}=2$
	2	$2\times1=2\rightarrow a_{ij}=1$	$a_{ij}=2$	$a_{ij}=2$	$a_{ij}=3$
	3	$3\times1=3\rightarrow a_{ij}=1$	$a_{ij}=2$	$a_{ij}=3$	$a_{ij}=3$
	4	$4\times1=4\rightarrow a_{ij}=2$	$a_{ij}=3$	$a_{ij}=3$	$a_{ij}=3$

k, o algoritmo identifica (k+1) *clusters* para os perfis de expressão e os perfis de condição, independentemente (GALLO *et al.*, 2015). Considerando uma matriz **E** de expressão gênica $M \times N$, onde M representa os valores de expressão e N os pontos no tempo, Bikmeans usa o *k-means*, que divide **E(m,:)**, de tal maneira que os valores de expressão adjacentes do gene m são divididos no mesmo intervalo. Além disso, Bikmeans usa Cokmeans (*k-means* para colunas), onde a matriz **E(:,n)** é dividida em k intervalos de tal maneira que os valores de expressão no tempo n são divididos no mesmo intervalo (LI *et al.*, 2010). Bikmeans computa (k+1)-*means clusters* para os valores de expressão e pontos do tempo, independentemente, dando a cada valor de expressão dois possíveis estados discretos, a'_{ij}: um para o valor de expressão (a^e_{ij}), e um para o ponto no tempo, a^t_{ij}, com $1 \leq a^e_{ij} \leq k+1$, $1 \leq a^t_{ij} \leq k+1$. Logo, o estado discreto a_{ij}, com $1 \leq a_{ij} \leq$ k é atribuído a a'_{ij} se $(a_{ij})^2 \leq a^g_{ij}a^c_{ij} < (a_{ij}+1)^2$. Considerando k=3, a Tabela 2 apresenta um exemplo dos possíveis estados discretos para a_{ij}. Nesse caso, *k-means* é executado M + N vezes, uma vez que tanto os valores de expressão quanto os pontos no tempo participam do agrupamento (GALLO *et al.*, 2015). Por fim, uma abordagem desenvolvida por (GALLO; CARBALLIDO; PONZONI, 2011), onde a discretização de um gene i pode ser definida como

$$\min_{S_1,S_2\subset S}(var(S_1)+var(S_2)), \tag{2.27}$$

onde S é o conjunto de valores da amostra do gene i, $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$, $|S_1| > 1$ and $|S_2| > 1$, var(S_1) e var(S_2) são as variâncias de S_1 e S_2, respectivamente, e S_1 e S_2 representam os dois estados para o gene i. As amostras do gene i são divididas em dois conjuntos que possuem a menor soma de suas variâncias. As cardinalidades de S_1 e S_2 devem ser maiores que um a fim de evitar efeitos de possíveis *outliers* nas amostras (GALLO; CARBALLIDO; PONZONI, 2011). Portanto, quando as amostras do gene i são separadas numa partição que viola essa restrição, o gene i não é mais considerado no processo de inferência corrente.

Uma abordagem de discretização diferente é considerar a variação entre os pontos do tempo ao invés dos valores absolutos de expressão gênica. Neste caso, os métodos são aplicáveis somente aos dados de expressão gênica na forma de séries temporais do mesmo experimento, computando assim como os perfis de expressão evoluem ao longo do tempo para realizar a discretização. Desta forma, segundo GALLO *et al.* (2015), os únicos

escopos de dados significantes são o perfil de expressão ou a variação no tempo.

Nesta categoria de abordagens de discretização, a abordagem mais simples é chamada de *Transitional State Discrimination (TSD)* (MÖLLER-LEVET; CHU; WOLKENHAUER, 2003). TSD inicialmente padroniza os valores utilizando z-*scores* com distribuição normal. Os estados discretos são então atribuídos conforme

$$a_{ij} = \begin{cases} 1, & a'_{ij} - a'_{i(j-1)} \geq 0, e \\ 0, & a'_{ij} - a'_{i(j-1)} < 0 \end{cases} \tag{2.28}$$

onde a_{ij} é o valor discretizado e a'_{ij} e $a'_{i(j-1)}$ são os dados reais. De acordo com o TSD, a_{ij} é igual a zero (inativo) quando os valores observados diminuem, e um (ativo), caso contrário. Um método relacionado ao TSD foi desenvolvido por ERDAL *et al.* (2004), no qual a diferença absoluta entre pontos sucessivos no tempo é aplicado. Esse método introduz um *threshold* ι para os estados discretos com regulação positiva, como

$$a_{ij} = \begin{cases} 1, & |a'_{ij} - a'_{i(j-1)}| \geq \iota, e \\ 0, & \text{caso contrário} \end{cases} \tag{2.29}$$

Considerando um nível de discretização de três, uma abordagem simples é combinar a discretização por média com as variações entre pontos no tempo (SOINOV; KRESTYANINOVA; BRAZMA, 2003; PONZONI *et al.*, 2007). Nesse caso, o primeiro passo é discretizar a matriz de expressão gênica A' utilizando os valores absolutos com a abordagem de discretização por média no perfil de expressão. Isso produz uma matriz discretizada intermediária A". Então, cada estado discreto é obtido segundo

$$a_{ij} = (a''_{ij} - a''_{i(j-1)}) \tag{2.30}$$

Essa abordagem fornece uma matriz discretizada A de N genes e M-1 condições, na qual para cada a_{ij} é atribuído um estado discreto: 1, -1 e 0, significando aumento, diminuição e sem mudança, respectivamente. Esse método foi aplicado em (SOINOV; KRESTYANINOVA; BRAZMA, 2003; PONZONI *et al.*, 2007).

Outra abordagem consiste em analisar as variações entre os pontos sucessivos do tempo mas considerando que essas variações só são significantes segundo um *threshold* (MADEIRA; OLIVEIRA, 2005; ZOMAYA; ELLOUMI, 2013; JI; TAN, 2004; MADEIRA *et al.*, 2008). Dessa forma, a matriz discretizada A pode ser obtida segundo dois passos principais: (i) primeiro transforma-se a matriz A' em uma matriz A" de variações, segundo a Equação 2.31, e (ii) a matriz discretizada final A é obtida considerando um *threshold* $\iota > 0$, como apresentado na Equação 2.32.

$$a''_{ij} = \begin{cases} \frac{a'_{ij} - a'_{i(j-1)}}{|a'_{i_{(j-1)}|}}, & \text{se } |a'_{i(j-1)} \neq 0 \\ 1, & \text{se } a'_{i(j-1)} = 0 \wedge a'_{ij} > 0 \\ -1 & \text{se } a'_{i(j-1)} = 0 \wedge a'_{ij} < 0 \\ 0, & \text{se } a'_{i(j-1)} = 0 \wedge a'_{ij} = 0 \end{cases} \tag{2.31}$$

$$a_{ij} = \begin{cases} 1, & \text{se } a''_{ij} \geq \iota \\ -1 & \text{se } a''_{ij} \leq -\iota \\ 0, & \text{caso contrário} \end{cases} \tag{2.32}$$

Diversos usos deste abordagem podem ser encontrados na literatura (MADEIRA; OLIVEIRA, 2005; ZOMAYA; ELLOUMI, 2013; JI; TAN, 2004; MADEIRA *et al.*, 2008).

Todos os métodos de discretização apresentados nesta seção podem ser utilizados a partir da ferramenta *Gene Expression Data Pre-Processing Tool* (GEDPROTOOLS) (GALLO *et al.*, 2015), disponível publicamente[6].

2.11.2 Conversão de um modelo discreto para contínuo

Conforme destacado anteriormente, modelos Booleanos constituem a forma mais simples de modelagem de GRNs, simplificando a estrutura e a dinâmica de regulação gênica. Esse tipo de modelo provê uma medida qualitativa dos mecanismos regulatórios gênicos. Por outro lado, modelos baseados em sistemas de EDOs são modelos de GRNs simbólicos acurados, capazes de representar quantitativamente as interações gênicas e possibilitando a predição dos níveis de expressão gênica ao longo do tempo. Desta forma, a conversão de um modelo Booleano funcional em um modelo contínuo expresso por um sistema de EDOs constitui um enriquecimento do significado do modelo e consequentemente das informações que podem ser extraídas.

A metodologia considerada aqui para obter um modelo contínuo a partir de modelos Booleanos é apresentada em (WITTMANN *et al.*, 2009). Em geral, um modelo Booleano consiste de N espécies, $X_1, X_2, \ldots, X_N$, e cada espécie toma valores em $x_i \in \{0, 1\}$. Além disso, existe uma função de atualização discreta B_i para cada espécie $X_i(t)$, no tempo t, que fornece seu valor no instante $t + 1$:

$$X_i(t+1) = B_i(x_{i1}(t), x_{i2}(t), ..., x_{iN_j}(t)) \in \{0, 1\} \tag{2.33}$$

O primeiro passo para obter um modelo contínuo a partir do modelo Booleano é transformar cada variável discreta x_i em uma variável contínua $\overline{x_i} \in [0, 1]$, onde as concentrações são normalizadas no intervalo unitário (WITTMANN *et al.*, 2009). Para tanto, B_i é transformada em uma função de atualização contínua $\overline{B_i}$. A função de atualização contínua ($\overline{B_i}$) é definida aqui utilizando HillCubes. Inicialmente, BooleCubes são obtidos a partir de uma interpolação polinomial multivariada, de acordo com:

$$\overline{B_i}(\overline{x_1}, ..., \overline{x_N}) = \sum_{x_1=0}^{1} ... \sum_{x_N=0}^{1} B(x_1, ..., x_N) \prod_{i=1}^{N} (x_i \overline{x_i} + (1 - x_i)(1 - \overline{x_i})) \tag{2.34}$$

onde B$(x_1, x_2, ..., x_N)$ representa a função de atualização da espécie X_i. A Equação 2.34 pode ser vista como a soma de produtos em domínio contínuo. A Tabela 3 exemplifica a

[6] `http://lidecc.cs.uns.edu.ar/files/gedprotools.zip`

Tabela 3 – Transformação de funções de atualização discretas em funções de atualização contínuas na forma de BooleCubes.

a	b	c	B	$\overline{B}$
0	0	0	0	0
0	0	1	1	$(1-\overline{x_a})\cdot(1-\overline{x_b})\cdot\overline{x_c}$
0	1	0	1	$(1-\overline{x_a})\cdot\overline{x_b}\cdot(1-\overline{x_c})$
0	1	1	0	0
1	0	0	0	0
1	0	1	1	$\overline{x_a}\cdot(1-\overline{x_b})\cdot\overline{x_c}$
1	1	0	1	$\overline{x_a}\cdot\overline{x_b}\cdot(1-\overline{x_c})$
1	1	1	1	$\overline{x_a}\cdot\overline{x_b}\cdot\overline{x_c}$

Fonte: Elaborado pelo autor (2022).

transformação da função de atualização discreta (B) para a função de atualização contínua ($\overline{B}$) na forma de BooleCube, para um circuito de três variáveis (A, B e C). As variáveis no domínio contínuo são representadas por $\overline{x_i}$ com $i \in \{a, b, c\}$ e o BooleCube final do exemplo é dado por:

$$\overline{B_i} = (1-\overline{x_a})\cdot(1-\overline{x_b})\cdot\overline{x_c}+(1-\overline{x_a})\cdot\overline{x_b}\cdot(1-\overline{x_c})+\overline{x_a}\cdot(1-\overline{x_b})\cdot\overline{x_c}+\overline{x_a}\cdot\overline{x_b}\cdot(1-\overline{x_c})+\overline{x_a}\cdot\overline{x_b}\cdot\overline{x_c} \quad (2.35)$$

Contudo, BooleCubes não conseguem representar o formado sigmoidal comumente presente em interações biológicas. Desta forma, faz-se necessário a transformação dos BooleCubes em HillCubes, uma vez que estes são capazes de representar esse comportamento de comutação, utilizando as funções sigmoidais de Hill. A função de Hill é definida por:

$$f(\overline{x}) = \frac{\overline{x}^z}{\overline{x}^z + k^z}, \quad (2.36)$$

onde z determina a inclinação da curva (cooperatividade de interação) e k é um *threshold*.

As variáveis contínuas $\overline{x_i}$ são substituídas pelas funções de Hill em $\overline{B_i}$ e a nova função de atualização contínua, chamada HillCube é definida como:

$$\overline{B}_i^H(\overline{x}_{i1}, ..., \overline{x}_{iN_i}) = \overline{B}_i\left(f_{i1}(\overline{x}_{i1}), ..., f_{iX_i}(\overline{x}_{iN_i})\right). \quad (2.37)$$

Considerando o exemplo da Tabela 3 e o BooleCube resultante da Equação 2.35, o HillCube é obtido aplicando-se a função de Hill apresentada na Equação 2.36 em cada variável contínua $\overline{x_i}$. O resultado é:

$$\begin{aligned}\overline{B_i^H} = &(1 - \frac{\overline{x_a}^{z_a}}{\overline{x_a}^{z_a} + k_a^{z_a}}) \cdot (1 - \frac{\overline{x_b}^{z_b}}{\overline{x_b}^{z_b} + k_b^{z_b}}) \cdot \frac{\overline{x_c}^{z_c}}{\overline{x_c}^{z_c} + k_c^{z_c}} + \\ &(1 - \frac{\overline{x_a}^{z_a}}{\overline{x_a}^{z_a} + k_a^{z_a}}) \cdot \frac{\overline{x_b}^{z_b}}{\overline{x_b}^{z_b} + k_b^{z_b}} \cdot (1 - \frac{\overline{x_c}^{z_c}}{\overline{x_c}^{z_c} + k_c^{z_c}}) + \\ &\frac{\overline{x_a}^{z_a}}{\overline{x_a}^{z_a} + k_a^{z_a}} \cdot (1 - \frac{\overline{x_b}^{z_b}}{\overline{x_b}^{z_b} + k_b^{z_b}}) \cdot \frac{\overline{x_c}^{z_c}}{\overline{x_c}^{z_c} + k_c^{z_c}} + \\ &\frac{\overline{x_a}^{z_a}}{\overline{x_a}^{z_a} + k_a^{z_a}} \cdot \frac{\overline{x_b}^{z_b}}{\overline{x_b}^{z_b} + k_b^{z_b}} \cdot (1 - \frac{\overline{x_c}^{z_c}}{\overline{x_c}^{z_c} + k_c^{z_c}}) + \\ &\frac{\overline{x_a}^{z_a}}{\overline{x_a}^{z_a} + k_a^{z_a}} \cdot \frac{\overline{x_b}^{z_b}}{\overline{x_b}^{z_b} + k_b^{z_b}} \cdot \frac{\overline{x_c}^{z_c}}{\overline{x_c}^{z_c} + k_c^{z_c}}\end{aligned} \tag{2.38}$$

Devido à formulação matemática, HillCubes nunca assumem valor 1. Para que isso seja possível, basta normalizar as funções de Hill, conforme apresentado na Equação 2.39.

$$\overline{B}_i^{Hn}(\overline{x}_{i1}, ..., \overline{x}_{iNi}) = \overline{B}_i^I \left(\frac{f_{i1}(\overline{x}_{i1})}{f_{i1}(1)}, ..., \frac{f_{iN_i}(\overline{x}_{iN_i})}{f_{iN_i}(1)} \right) \tag{2.39}$$

Por fim, o comportamento temporal de $\overline{x}_i$ é obtido aplicando-se a função de atualização contínua em:

$$\dot{\overline{x_i}} = \frac{1}{\tau_i}(\overline{B}(\overline{x_{i1}}, \overline{x_{i2}}, ..., \overline{x_{iN_i}}) - \overline{x_i}) \tag{2.40}$$

Esse comportamento é composto por:

1. a função de atualização contínua $\overline{B}$, que escreve a produção das espécies X_i e um termo de decaímento de primeira ordem, e
2. seu parâmetro correspondente τ_i que pode ser entendido como o tempo de vida das espécies X_i.

Portanto, de maneira resumida, um sistema de EDOs é determinado utilizando HillCubes através dos seguintes passos:

1. obter um BooleCube através de uma interpolação polinomial multivariada, apresentada na Equação 2.34, e
2. transformar o BooleCube em um HillCube.

O modelo gerado contém coeficientes numéricos (z, k e τ) que precisam ser determinados. É importante ressaltar que a Tabela 3 apresenta uma única saída e, portanto, os parâmetros n e k da função de Hill dizem respeito à essa saída. No caso de múltiplas saídas, cada uma das saídas terá seus parâmetros z e k. Pode-se observar que o tempo de vida de cada espécie (τ) também é um parâmetro dependente da saída em questão.

2.12 MÉTRICAS DE AVALIAÇÃO E REDES DE REFERÊNCIA

Avaliar uma GRN inferida é um processo complexo. Isso se deve ao fato da ausência de conjuntos de problemas padronizados e a inexistência da rede original (*ground-truth network*) (PRATAPA *et al.*, 2020; TIAN *et al.*, 2019).

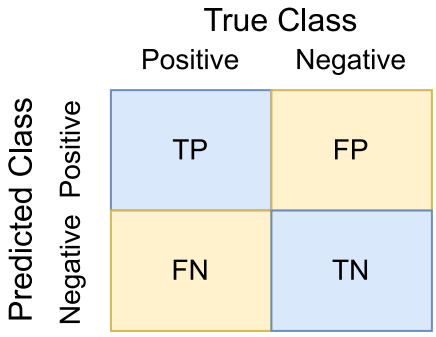
Diferentemente do que é comumente apresentado na área de aprendizado de máquina, a avaliação de GRNs e redes de coexpressão, quando realizadas de maneira qualitativa, partem do princípio da comparação entre o que foi obtido pelo método de inferência e uma dada rede. Por este motivo, torna-se inviável a divisão entre dados de treino e teste ou treino, validação e teste. Contudo, para modelos contínuos de GRNs, onde consideram-se os níveis de expressão gênica, concentrações de mRNA, entre outros, é possível separar os dados em conjuntos de treino e teste apropriados. Nesse caso, o resultado obtido pelo modelo é avaliado utilizando um conjunto de dados separado daquele usado durante o treinamento.

Tendo em vista estes aspectos, foi proposto em (PRATAPA *et al.*, 2020) um conjunto de problemas *benchmark* e um *framework* denominado BEELINE, que contém uma série de métricas para a avaliação das redes inferidas.

Antes de iniciar a discussão sobre as métricas comumente utilizadas na literatura para atribuir qualidade as GRNs inferidas, CANBEK *et al.* (2017) define métricas básicas e aquelas obtidas a partir da matriz de confusão.

Uma matriz de confusão é uma matriz $N \times N$ usada para avaliar o desempenho do modelo de classificação, onde N é o número total de classes alvo. A matriz compara os valores alvo atuais com aqueles preditos pelo modelo. Isso fornece uma visão holística do desempenho e do tipo de erros de um modelo de classificação. A representação de uma matriz de confusão é apresentada na Figura 26.

Figura 26 – Matriz de Confusão.



Fonte: Elaborado pelo autor (2024).

Na Figura 26, TP, TN, FP e FN são *True Positive* (Verdadeiros Positivos), *True Negative* (Verdadeiros Negativos), *False Positive* (Falsos Positivos) e *False Negative* (Falsos Negativos), respectivamente. Um Verdadeiro Positivo (TP) é um resultado onde o modelo

prediz corretamente a classe positiva. De maneira análoga, um Verdadeiro Negativo (TN) é um resultado corretamente predito pelo modelo, da classe negativa. Um Falso Positivo (FP) é um resultado em que o modelo prediz incorretamente a classe positiva e, por fim, um Falso Negativo (FN) é um resultado onde o modelo prediz incorretamente a classe negativa. Essas quatro medidas básicas são obtidas diretamente da matriz de confusão. O resultado da classificação verdadeira, ou correspondências de previsão/realidade (TP e TN) estão na diagonal principal. As não correspondências (FP e FN), que representam resultados falsos, estão fora da diagonal principal.

Quatro métricas podem ser extraídas a partir dessas medidas básicas, baseadas nas colunas ou linhas da matriz. A métricas básicas do tipo coluna são *True Positive Rate* (TPR), *True Negative Rate* (TNR), *False Positive Rate* (FPR), e *False Negative Rate* (FNR). Já as métricas básicas do tipo linha são *Positive Predictive Value* (PPV), *Negative Predictive Value* (NPV), *False Discovery Rate* (FDR), e *False Omission Rate* (FOR). Essas métricas de primeiro nível são as métricas mais conhecidas e preferíveis para expressar o desempenho da classificação binária através de um valor único (CANBEK *et al.*, 2017).

O TPR, também chamado de *sensitivity* ou *recall* é a razão de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos. Este resultado é a probabilidade de um resultado de teste verdadeiramente positivo, condicionado à amostra ser verdadeiramente positiva.

De maneira similar, levando em consideração os verdadeiros negativos e os falsos positivos, temos o TNR, também chamado de especificidade. Seu significado é a probabilidade de um resultado de teste verdadeiramente negativo, condicionado à amostra ser verdadeiramente negativa. O FPR e o FNR são similares aos anteriores, mas consideram o FP e o FN, respectivamente.

PPV, também conhecido como precisão, é a fração de verdadeiros positivos dentre as instâncias classificadas como positivas. Já o NPV, é a fração de verdadeiros negativos dentre as instâncias classificadas como negativas. Por fim, o FRD e o FOR são similares ao PPV e NPV, mas considerando a fração de falsos positivos dentre as instâncias classificadas como positiva e a fração de falsos negativos dentre as instâncias classificadas como negativas, respectivamente.

Diversas outras métricas são derivadas a partir dessas métricas base. Por exemplo, *accuracy* (ACC) e *F-Score*.

ACC é definida como a fração de valores verdadeiros entre todos os valores. Contudo, essa métrica fornecer informações enviesadas quando os dados são desbalanceados. Por esse motivo, uma maneira mais equitativa de calcular acurácia é através da *balanced accuracy* (BACC), que considera a média de TPR e TNR. Por fim, *F-Score* é calculado a partir da *precision* e *recall*. A forma mais geral é dada por $F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$.

Assumindo $\beta = 1$, temos $F_1 = 2\frac{precision \cdot recall}{precision + recall}$, e significa a média harmônica de *precision* e *recall*. A Tabela 4 resume as métricas e suas respectivas fórmulas.

Tabela 4 – Sumário das métricas e suas fórmulas.

Metric	Formula
TPR Sensitivity Recall	$\frac{TP}{TP+FN}$
TNR Specificity Selectivity	$\frac{TN}{TN+FP}$
FPR	$\frac{FP}{FP+TN}$
FNR	$\frac{FN}{FN+TP}$
PPV Precision	$\frac{TP}{TP+FP}$
NPV	$\frac{TN}{TN+FN}$
FDR	$\frac{FP}{FP+TP}$
FOR	$\frac{FN}{FN+TN}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+TN}$
Balanced Accuracy	$\frac{TPR+TNR}{2}$
F_β-Score	$(1+\beta^2)\frac{Precision \times Recall}{(\beta^2 \times Precision)+Recall}$

Fonte: Elaborado pelo autor (2024).

Para maiores informações sobre métricas e medidas para classificadores binários, refere-se a (CANBEK *et al.*, 2017).

No trabalho de PRATAPA *et al.* (2020), três métricas são consideradas: área sob a curva de *precision-recall* (AUPRC - do inglês *area under the precision-recall curve*), a área sob a curva ROC (AUROC - do inglês *area under the receiver operating characteristic curve*), a *early precision* (EP), e *early precision ratio* (EPR).

AUPRC é calculada como a área sob a curva de *precision-recall* (PR) e mostra o *trade-off* entre a precisão e a revocação em diferentes limites de decisão. O eixo x da curva PR é a revocação e o eixo y é a precisão. Pode-se definir a precisão como $\frac{TP}{TP+FP}$ e a revocação como $\frac{TP}{TP+FN}$.

A curva ROC por sua vez é obtida considerando a razão de falsos positivos (FPR) no eixo x e o *recall* no eixo y. O FPR é calculado como $\frac{FP}{TN+FP}$. Um modelo bom tem AUC, tanto para ROC quanto para PRC, próximos a 1. Enquanto AUROC = 1 significa que existe boa separabilidade, mas não necessariamente qualidade, AUPRC = 1 indica a não existência de erros de predição. Para muitos conjuntos de dados reais, particularmente

conjuntos de dados médicos, a fração de positivos é geralmente menor que 0,5, o que significa que AUPRC tem um valor de linha de base menor que AUROC.

Por fim, EP é definido por PRATAPA *et al.* (2020) como a fração de verdadeiros positivos nas *top-k* arestas, onde k é o número de arestas presentes na rede de referência, excluindo-se os auto *loops*. O EPR por sua vez é definido como o EP do método a ser avaliado dividido pelo EP de um preditor aleatório para aquele modelo. A precisão do preditor aleatório é a densidade das arestas da rede de referência, definida como $\frac{nReg}{(nTFs \times nGenes) - nTFs}$, onde $nReg$ é o número de relações regulatórias da rede de referência e $nTFs$ e $nGenes$ são os números de fatores de transcrição e número de genes presentes no conjunto de dados, respectivamente. O uso desta métrica é justificado por PRATAPA *et al.* (2020) devido ao fato de que as interações preditas de maior confiança serão mais interessantes para os experimentalistas.

Para realizar o cálculo dessas métricas, faz-se necessário comparar os resultados obtidos com uma rede de referência. Em alguns casos a rede é conhecida (*ground-truth*). Contudo, em casos práticos, tal rede é desconhecida. Por esse motivo, é comum utilizar redes de referência. Neste trabalho são consideradas três redes de referência, conforme apresentado em (PRATAPA *et al.*, 2020): STRING, NonSpecific ChIP-Seq e *cell-type-specific* ChIP-Seq.

A rede STRING representa interações que são derivadas de evidência baseada em resultados experimentais, bancos de dados de vias metabólicas ou de transdução de sinal, mineração de texto, e outras fontes de informação SZKLARCZYK *et al.* (2015). É importante ressaltar que as relações apresentadas na rede STRING são funcionais e não correspondem, necessariamente, a regulação transcricional (PRATAPA *et al.*, 2020). De acordo com PRATAPA *et al.* (2020), o uso da rede STRING é justificado uma vez que alguns métodos de inferência de modelos Booleanos podem predizer relações indiretas.

As redes NonSpecific ChIP-Seq são uma coleção de relações regulatórias obtidas de diferentes fontes que não levam em consideração o tipo celulas específico. Diversos recursos tais como DoRothEA (GARCIA-ALONSO *et al.*, 2019), que integra informação regulatória transcricional de múltiplas fontes, RegNetwork (LIU *et al.*, 2015) que incorpora relações regulatórias de genoma inteiro TF-TF, TF-gene, e TF-microRNA, e TRRUST (HAN *et al.*, 2018) contendo interações TF-alvo baseadas em mineração de texto e curadoria manual são utilizados para gerar redes NonSpecific.

Por fim, as redes *cell-type-specific* ChIP-Seq são saídas de GRNs transcricionais que conectam fatores de transcrição e proteínas de sinais a genes alvo (ZHANG *et al.*, 2023). Essas redes são construídas considerando conjuntos de dados para dados ChIP-Seq do mesmo tipo celular ou de tipos celulares similares, tais como ENCODE[7], ChIP-Atlas[8],

[7] `https://www.encodeproject.org/data-standards/chip-seq/`
[8] `https://chip-atlas.org/`

e ESCAPE[9].

Além disso, neste trabalho, são utilizados os perfis de desempenho (PPs - do inglês *performance profiles*) (DOLAN; MORÉ, 2002; BARBOSA; BERNARDINO; BARRETO, 2010) para analisar o desempenho relativo dos algoritmos. Com um conjunto de S resolvedores (algoritmos) s_i, $i \in \{1, \ldots, n_s\}$, e o conjunto P de problemas p_j, $j \in \{1, \ldots, n_p\}$, $t_{p,s}$ pode ser definido como medida de desempenho para o método $s \in S$ quando resolvendo o problema $p \in P$. A razão de desempenho $r_{p,s} = \frac{t_{p,s}}{\max\{t_{p,s}:s\in S\}}$ é o desempenho relativo do método s. Usando a razão de desempenho $r_{p,s}$, a probabilidade que a razão de desempenho $r_{p,s}$ do método $s \in S$ está dentro de um fator $\tau > 0$ da melhor relação possível pode ser definida como $\rho_s(\tau) = \frac{1}{n_p} |\{p \in P : r_{p,s} \leq \tau\}|$, onde $\rho_s(\tau)$ denota as curvas dos PPs. A partir dos PPs, é possível extrair:

1. a abordagem que obteve os melhores resultados para a maioria dos problemas (maior $\rho(1)$),
2. a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e
3. o melhor desempenho geral (maior área sobre as curvas dos PPs).

Testes estatísticos também são realizados para determinar se os resultados são originados de uma mesma distribuição. Para esse fim, neste trabalho, utiliza-se o teste de Kruskal-Wallis. O teste de Kruskal-Wallis estende o teste U de Mann-Whitney para mais de dois grupos. A hipótese nula do teste de Kruskal-Wallis é que as classificações médias dos grupos são iguais. Além disso, o teste não paramétrico de Kruskal-Wallis não assume uma distribuição normal dos dados subjacentes. Como teste *post-hoc*, utiliza-se o teste não paramétrico de Dunn. Quando o p-valor $\leq 0,05$, conclui-se que a hipótese nula é rejeitada.

Como será discutido na Seção 2.13, GRNs podem ser entendidas como classificadores binários. Por esse motivo, e tendo em vista a natureza de dados desbalanceados, PUŠNIK *et al.* (2022) utilizam o *Matthews correlation coefficient* (MCC). O MCC é uma medida de associação para duas variáveis binárias e é comumente utilizada para medir a qualidade de classificadores binários. O MCC pode ser calculado diretamente a partir da matriz de confusão através da fórmula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.41}$$

[9] `http://www.maayanlab.net/ESCAPE/`

2.13 PROCEDIMENTO METODOLÓGICO EM SCRNA-SEQ

Conforme discutido ao longo deste trabalho, diversos aspectos, tais como o tipo de tecnologia de dados, as redes de referência e o desconhecimento da rede *ground-truth*, as métricas de avaliação, os genes que serão utilizados, a modelagem da informação temporal e a etapa de pré-processamento, e os *motifs* de rede influenciam no processo de inferência e avaliação de GRNs.

Em relação à tecnologia de sequenciamento de dados, em específico para o scRNA-Seq, a maioria dos níveis de expressão gênica são reportados como zero. Esse efeito, conhecido como *dropout*, tornam a análise dos dados de expressão gênica uma tarefa complexa, tendo em vista a dificuldade em determinar a fonte dos zeros observados nos dados e constitui um dos principais desafios na análise computacional (ANDREWS *et al.*, 2020). Além disso, a variação biológica, tais como a natureza estocástica da expressão gênica, o nicho de ambiente e efeitos criados pelo ciclo celular podem introduzir erros nos dados (HWANG; LEE; BANG, 2018). Contudo, de maneira geral, a maioria dos métodos de inferência de GRNs partem do pressuposto que o conjunto de dados de expressão gênica já passaram por testes de controle de qualidade associados ao processo de sequenciamento. Por outro lado, conforme ressaltado na literatura, os métodos de inferência de GRN, específicos ou não para scRNA-Seq, tendem a apresentar comportamento próximo a preditores aleatórios (PRATAPA *et al.*, 2020; CHEN; MAR, 2018).

A maioria dos organismos possui ciclos de realimentação e sistemas de autorregulação em seus circuitos de transcrição, conforme discutido na Seção **2.2.4**. Contudo, apesar de tal *motif* de rede ser importante para os processos de manutenção de vida em diversos organismos, é comum encontrar na literatura trabalhos que removem a autorregulação para a avaliação (CHEN; MAR, 2018; PRATAPA *et al.*, 2020). A principal justificativa reside no fato de que muitos algoritmos tendem a priorizar a obtenção de tais relações regulatórias enquanto outros tendem a sempre ignorá-la (PRATAPA *et al.*, 2020).

Uma outra discussão diz respeito ao uso de informação temporal/pseudotemporal para a inferência de GRNs. Conforme ressaltado por PRATAPA *et al.* (2020), algoritmos que necessitam de tal informação apresentaram resultados inferiores em relação aqueles que não necessitam. Contudo, mostramos na Seção 4 que o CGPGRN apresenta resultados melhores ou competitivos com os resultados obtidos pelos algoritmos estado da arte.

Como discutido na Seção 3.2, nem todos os genes estão expressos o tempo todo. Consequentemente, nem todos os genes estão associados a todos os fenômenos biológicos. Por esse motivo, determinar corretamente o subconjunto de genes mais representativo para modelar o fenômeno biológico de interesse torna-se uma tarefa importante (YANG; HUANG; LIU, 2021).

Medir a qualidade de uma GRN inferida também é um processo complexo (ZHAO

et al., 2021; MARBACH *et al.*, 2010; AKERS; MURALI, 2021; PRATAPA *et al.*, 2020; CHEN; MAR, 2018). Isso deve-se principalmente a dois fatos: o desconhecimento das redes *ground-truth* e o conjunto de métricas utilizado para medir a qualidade. Em situações reais, não existe conhecimento da rede *ground-truth* e os algoritmos de inferência são utilizados para auxiliar numa primeira análise dos dados de expressão gênica. Com isso, os experimentalistas podem utilizar tais informações para realizar experimentos adicionais. Além disso, como discutido na Seção 2.12, uma vez que os métodos de inferência de GRNs podem ser entendidos como classificadores binários e que os dados de expressão gênica são tipicamente desbalanceados, utilizar um conjunto de métricas apropriados para essa classe de problemas torna-se fundamental para a interpretação correta dos resultados obtidos.

Nesta seção são apresentados os processos metodológicos comumente adotados na literatura, relacionados à modelagem e inferência de GRNs utilizando a tecnologia de perfilamento scRNA-Seq.

2.13.1 Seleção de Subconjuntos de Genes

Selecionar um subconjunto de genes é tratado como seleção de características em bioinformática. Os avanços recentes nas tecnologias de *single-cell* resultaram em conjuntos de dados de alta dimensionalidade com complexidade aumentada, tornando a seleção de características uma técnica essencial para a análise de dados de célula única (YANG; HUANG; LIU, 2021).

O termo seleção de características refere-se a uma classe de métodos computacionais que selecionam um subconjunto de características úteis a partir das características originais (YANG; HUANG; LIU, 2021). Contudo, é importante distinguir seleção de características de redução de dimensionalidade. Redução de dimensionalidade consiste em combinar e/ou transformar dados para derivar-se uma dimensão de características menor. Já a seleção de características pode ser entendida como uma redução de dimensionalidade mas mantendo o espaço de características originais intactos (HAUSKRECHT *et al.*, 2007; YANG; HUANG; LIU, 2021).

De acordo com SAEYS; INZA; LARRANAGA (2007), a aplicação de seleção de características em bioinformática é amplo. Alguma das direções populares de pesquisa incluem a seleção de genes que podem discriminar doenças complexas, como o câncer (LAZAR *et al.*, 2012; BOLÓN-CANEDO *et al.*, 2014). Para um *survey* completo em métodos de seleção de características, referencia-se (CHANDRASHEKAR; SAHIN, 2014). No contexto de aplicações em bioinformática, referencia-se (YANG; HUANG; LIU, 2021). Tradicionalmente, de acordo com YANG; HUANG; LIU (2021), as técnicas de seleção de características podem ser de três tipos: filtros, *wrappers* e métodos incorporados.

O conceito subjacente aos métodos filtro é a ordenação das características baseadas em um certo critério, tal como um *threshold*, para facilitar as análises subsequentes. Isso é

aplicado, por exemplo, na discriminação de amostras. Em aplicações de bioinformática, métodos de análise univariada são comumente utilizadas. Alguns exemplos incluem a estatística t, na qual a maioria dos métodos de expressão diferencial (DE - do inglês, *Differential Expression*) para análise de dados biológicos são construídos (RITCHIE *et al.*, 2015). As principais vantagens dos métodos filtro residem em sua simplicidade, tendo em vista a necessidade de menos recursos computacionais e a facilidade de aplicação na prática (BOMMERT *et al.*, 2020).

Wrappers utilizam o desempenho dos algoritmos de indução para guiar o processo de seleção de características e, portanto, podem levar a características que são mais propícias para o algoritmo de indução utilizado para otimização nas análises posteriores (KOHAVI; JOHN, 1997). Um aspecto chave dos métodos *wrapper* é o projeto dos algoritmos de otimização de características que otimizam o desempenho do algoritmo de indução (YANG; HUANG; LIU, 2021). Para tal, diversos algoritmos gulosos, tais como o algoritmo genético (GA - do inglês *genetic algorithm*) (LI *et al.*, 2001) e a otimização por enxame de partículas (PSO - do inglês *particle swarm optimization*) (YANG *et al.*, 2009) foram utilizados para acelerar o processo de otimização e seleção de características.

Por fim, métodos incorporados realizam a seleção e a indução simultaneamente. Isso pode levar a características mais adequadas para o algoritmo de indução nas tarefas subsequentes, como a classificação de amostras (YANG; HUANG; LIU, 2021). Além disso, de acordo com BOLÓN-CANEDO; SÁNCHEZ-MAROÑO; ALONSO-BETANZOS (2013), tais métodos são geralmente mais eficientes computacionalmente que os métodos *wrapper*. Em aplicações de bioinformática, as escolhas mais comum residem nos métodos baseados em árvore (DENG; RUNGER, 2012; BREIMAN, 2001).

Quando considerando scRNA-Seq, de acordo com YANG; HUANG; LIU (2021), a seleção de características é amplamente aplicada na transcriptômica. Nesse caso, deseja-se selecionar genes a partir de dados scRNA-Seq para as análises a posteriores e pré-processamento. Alguns dos métodos mais populares são os filtros univariados, desenvolvidos para identificar genes diferencialmente distribuídos. Para tal, a estatística t ou métodos de expressão diferencial baseados em ANOVA (SONESON; ROBINSON, 2018; VANS; PATIL; SHARMA, 2021), além outras abordagens estatísticas tais como a variabilidade diferencial (DV - do inglês, *differential variability*) (LIN *et al.*, 2020) e a proporção diferencial (DP, do inglês, *differential proportion*) (KORTHAUER *et al.*, 2016) são utilizados para selecionar subconjuntos de genes.

Métodos baseados em distribuição diferencial podem frequentemente identificar genes que são altamente discriminativos para análises iniciais (YANG; HUANG; LIU, 2021). Contudo, tais métodos requerem rótulos definidos previamente, como tipos celulares. Esse requisito limita sua aplicabilidade quando essa informação não está disponível. Como alternativa, é possível filtrar por genes altamente variantes (HVG - do inglês, *highly variable*

genes), uma vez que tais métodos dependem somente da estimativa de variâncias. De acordo com YIP; SHAM; WANG (2019), métodos de detecção de HVG são comumente compostos por dois principais componentes: normalização e análise de variação. Como efeitos de lote podem efetivamente afetar o número de HVGs detectados (FINAK *et al.*, 2015), a normalização é uma etapa importante na análise de HVGs. Além disso, de acordo com SU; YU; WU (2021), diversos algoritmos de agrupamento para scRNA-Seq também implementam técnicas de identificação de HVGs e suas variantes para filtragem de genes a fim de melhorar o agrupamento das células. Um algoritmo comum para agrupamento é o k-means, conforme discutido na Seção 2.9. Para uma revisão de métodos de identificação de HVGs, referenciamos (YIP; SHAM; WANG, 2019).

Por fim, é importante ressaltar que um aspecto chave na aplicabilidade de métodos de seleção de características está na escalabilidade para grandes conjuntos de dados. Algoritmos de filtros univariados são os mais eficientes em termos de escalabilidade em relação à dimensão das características. Em geral, o tempo de processamento destes algoritmos cresce linearmente com o número de características (YANG; HUANG; LIU, 2021).

2.13.2 Pré-Processamento, *Motifs* de Rede e Inferência

Conforme discutido na Seção **2.2.5**, diversas tecnologias de perfilamento transcricional, tais como *microarrays*, RNA-Seq e scRNA-Seq, podem ser utilizadas. Enquanto dados oriundos de tecnologia *microarray* são capazes de quantificar de centenas a milhares de transcritos de uma dada célula ou amostra de tecido, RNA-Seq utiliza tecnologia de sequenciamento de próxima geração para estudar o transcriptoma como um todo. Contudo, o advento da tecnologia scRNA-Seq proveu oportunidades de explorar a expressão gênica a nível celular. Isso faz com que tal tecnologia seja favorável para estudar problemas biológicos centrais, tais como a heterogeneidade celular, a descoberta de novas subpopulações de células importantes durante a diferenciação celular, entre outros (CHEN; MAR, 2018; CHEN; NING; SHI, 2019).

Como não existe tempo físico nos experimentos scRNA-Seq, métodos de inferência de *pseudotimes* são utilizados a fim de fornecer tal informação. Estes métodos, por sua vez, ordenam as células ao longo de trajetórias que descrevem o desenvolvimento ou o progresso da célula (HAGHVERDI *et al.*, 2016; QIU *et al.*, 2017). Isso é particularmente útil em processos de diferenciação celular, tendo em vista que, a partir de uma dada célula inicial, diferentes *steady states* podem ser obtidos.

Contudo, como não existe método *gold-standard* para a inferência de pseudotimes, também não existe garantia de que os valores estimados das trajetórias representem a realidade. De acordo com PRATAPA *et al.* (2020), os algoritmos que requerem informação temporal apresentam desempenho inferior em relação aqueles que não necessitam dessa

informação. Entretanto, mostrou-se no decorrer desta tese que o CGPGRN, mesmo necessitando da informação temporal, é capaz de superar os algoritmos estado da arte na maioria dos casos.

Um aspecto chave é como lidar com os *pseudotimes* quando existem múltiplas trajetórias. Nesse caso, é possível inferir uma GRN independentemente para cada trajetória existente e unificar as redes obtidas para cada *pseudotime* ou usar todos os dados para inferir uma única GRN. Essas duas estratégias foram avaliadas em problemas sintéticos (PRATAPA *et al.*, 2020). Os resultados indicam que alguns algoritmos apresentam melhor desempenho quando inferindo uma única GRN, enquanto outros apresentam melhor desempenho inferindo GRNs para cada trajetória. De acordo com PRATAPA *et al.* (2020), essa ausência de benefícios claros entre separar as trajetórias ou não pode ser consequência da grande quantidade de interações do tipo falso positivo nos resultados, ressaltando novamente a característica de dados tipicamente desbalanceados. Entretanto, de acordo com o nosso conhecimento, a inferência de GRNs independentes por trajetória tende a aumentar a qualidade das redes inferidas, além de fornecer maiores informações sobre o fenômeno biológico que está sendo modelado (SILVA *et al.*, 2023).

Dessa forma, é importante que o experimentalista tenha objetivos claros a respeito das informações que deseja obter a partir das GRNs inferidas. A inferência de GRNs distintas por trajetória pode ser mais informativa quando deseja-se entender os processos de diferenciação celular de um tipo celular específico. Contudo, se o foco é identificar os genes mais significativos durante o processo de diferenciação, usar todas as trajetórias em uma única rede pode auxiliar em ressaltar aqueles genes mais importantes para a regulação como um todo.

Além da modelagem de informação temporal, alguns fatores podem afetar a confiabilidade e veracidade das informações que podem ser obtidas a partir do perfilamento scRNA-Seq (HWANG; LEE; BANG, 2018). Esses fatores podem ser principalmente de dois tipos: biológicos e técnicos. Os fatores biológicos residem no fato de que o RNA pode não ser expresso por um gene no momento da medição, na natureza estocástica da expressão gênica e nos efeitos criados pelo ciclo celular (HWANG; LEE; BANG, 2018). Já os fatores técnicos são ocasionados majoritariamente pela ineficiência de captura de mRNA (ANDREWS *et al.*, 2020). Dessa forma, a matriz de expressão gênica comumente contém muitos níveis de expressão reportados como zero. Isso dificulta distinguir e modelar apropriadamente as fontes de zeros observados e constitui um dos principais desafios para a análise computacional (ANDREWS *et al.*, 2020).

Como discutido na Seção 2.11, diversos algoritmos podem ser utilizados para a inferência de GRNs. Tais algoritmos diferenciam-se não só nas premissas adotadas, mas também em suas formulações matemáticas. Por exemplo, alguns algoritmos necessitam utilizar a informação temporal e outros não. Além disso, a maioria desses algoritmos

utilizam como entrada os dados brutos ou assumem que os dados fornecidos como entrada já estão prontos para uso. Contudo, devido às características próprias da tecnologia de sequenciamento por scRNA-Seq, faz-se necessária uma etapa de pré-processamento apropriada. Somente após isso, passos adicionais, tais como discretização dos dados para os modelos Booleanos, podem ser aplicados. Especificamente sobre discretização, conforme discutido nas Seções **2.11.1** e **4.6.3**, a escolha de um método de discretização pode afetar significativamente os resultados. Isso foi apresentado e discutido na Seção **4.6.1** onde mostrou-se que a etapa de pré-processamento adotada pelo CGPGRN é benéfica e aumenta a qualidade das GRNs inferidas.

Na Seção 2.2 foram discutidos diversos fatores que influenciam na expressão e regulação gênica. Em específico, na Seção **2.2.4** foram apresentados diversos *motifs* de rede que são conhecidos pela biologia.

Os *loops* de realimentação desempenham papéis essenciais no governo da dinâmica funcional das células. A importância desses *loops* em controlar a taxa de diferenciação celular e a progressão da linhagem celular em várias etapas já foi destacada (NORDICK; HONG, 2021). Por exemplo, *loops* de realimentação positivos geram uma memória de decisão celular em resposta a sinais transientes (histerese) (YAO *et al.*, 2008), enquanto *loops* de realimentação negativos produzem respostas adaptativas ou oscilatórias (MA *et al.*, 2009; POKHILKO *et al.*, 2012). Estudos teóricos e experimentais mais recentes revelaram que sistemas com mais de dois *loops* de realimentação interconectados têm funções adicionais no controle da dinâmica celular NORDICK; HONG (2021).

Existem também diversos indicativos de que os *self-loops*, ou autorregulação, são importantes em diferentes organismos, tais como no ciclo celular da *Budding Yeast* (KINOSHITA; YAMADA, 2018), e que esse *motif* pode oferecer vantages evolutivas nos mecanismos celuares (MONTAGNA; BRACCINI; ROLI, 2020). Outras discussões neste tópico estão nos raros estados transitórios de alta expressão coordenados em Câncer (SCHUH *et al.*, 2020), na existência de autorregulações nas interações regulatórias TF-TF (SINHA *et al.*, 2020), em ajudar a reproduzir fenômenos de diferenciação em redes Booleanas (BRACCINI; MONTAGNA; ROLI, 2019), e na flexibilidade conferida às chaves bioquímicas (PFEUTY; KANEKO, 2009).

Apesar de todas essas indicações sobre a importância de modelar corretamente as autorregulações, a literatura geralmente assume não considerar este tipo de regulação na avaliação de redes (CHEN; MAR, 2018; PRATAPA *et al.*, 2020). Uma justificativa comum é que alguns métodos de inferência sempre atribuem valores elevados às autorregulações, enquanto outras técnicas os ignoram (PRATAPA *et al.*, 2020).

Portanto, o desenvolvimento ou a escolha de um algoritmo de inferência de GRNs deve levar em consideração as especificidades dos dados que serão utilizados. Especificamente para scRNA-Seq, é importante modelar apropriadamente a informação temporal, a

autorregulação e outros *motifs* de rede, os erros biológicos e técnicos que podem ser introduzidos e as necessidades específicas do tipo de modelo adotado, tal como a discretização no caso de modelos Booleanos, tendo em vista seu impacto na identificação de relações regulatórias.

2.13.3 Avaliação e Redes de Referência

A avaliação de uma rede inferida é um processo complexo. Em geral, a avaliação de desempenho dos métodos de inferência é realizado através da comparação da GRN inferida com uma dada rede. Para dados sintéticos, é comum a existência de redes *ground-truth*. Contudo, para problemas experimentais, tal informação não está disponível. Nesse sentido, é importante discutir tanto os problemas utilizados para avaliar o desempenho dos métodos de inferência, bem como a rede de referência utilizada na comparação. Parte dessa discussão foi iniciada na Seção 2.12.

Quando lida-se com dados sintéticos, uma ferramenta amplamente utilizada para a geração destes dados é o GeneNetWeaver (SCHAFFTER; MARBACH; FLOREANO, 2011). Esse *software* é uma ferramenta *open-source* para a geração de dados *in silico*. Através do uso de redes transcricionais conhecidas (*E. coli*, *S. cerevisiae*, etc.), a estrutura das subredes são extraídas para obter as GRNs *in silico*. Então, modelos de dinâmica detalhados da regulação gênica são construídos, levando em consideração tanto a transcrição quanto a tradução, interações independentes e sinérgicas, bem como ruído molecular e de medição. Como resultado, dados de expressão, *steady-state* ou na forma de série temporais, para uma variedade de experimentos biológicos tais como *wild-type*, *knockout*, *knockdown* e experimentos multifatoriais podem ser obtidos. Além disso, GeneNetWeaver é utilizado para avaliar o desempenho de métodos de inferência nos desafios DREAM, um desafio anual de inferência de rede em toda a comunidade[10]. Para maiores informações sobre o GeneNetWeaver, refere-se a (SCHAFFTER; MARBACH; FLOREANO, 2011).

Outra possibilidade é o SyNtReN (BULCKE *et al.*, 2006). Essa ferramenta é um gerador de redes de regulação transcricional e produz dados de expressão gênica simulados. Esses dados simulados são uma aproximação dos dados experimentais. Dada uma rede de regulação previamente descrita, as topologias das redes são geradas através da seleção de subredes da rede original. As cinéticas de interação são modeladas por equações baseadas nas cinéticas de Michaelis-Menten e Hill. O SyNtReN já disponibiliza redes para *E. coli* e *S. cerevisiae* nativamente. Além disso, diversos parâmetros podem ser definidos pelo usuário a fim de ajustar a complexidade do *dataset* resultante, tais como ruídos biológicos e experimentais e a probabilidade de interações de regulação complexa entre dois reguladores.

Mais recentemente, PRATAPA *et al.* (2020) desenvolveram uma abordagem de-

[10] `https://dreamchallenges.org/`

nominada BoolODE[11]. Nessa ferramenta, para cada gene em uma GRN, uma função Booleana que especifica como os reguladores de tal gene combinam-se para controlar seu estado é requerida. Cada função Booleana é representada como uma tabela verdade e posteriormente convertida em uma equação diferencial ordinária não linear. Por fim, adiciona-se ruído a fim de tornar a equação estocástica. BoolODE permite configurar o número de células desejadas e a taxa de *dropout* para cada um dos *datasets* gerados. Conforme discutido na Seção 4.1, essa ferramenta foi utilizada para gerar os modelos sintéticos e acurados.

Como apresentado na Seção 4.1, PRATAPA *et al.* (2020) consideram o uso de problemas sintéticos, acurados e experimentais. Quando analisam-se os resultados de diversos algoritmos para a inferência de GRNs utilizando dados sintéticos, a maioria das técnicas apresentaram bom desempenho. Para dados sintéticos, os melhores algoritmos segundo PRATAPA *et al.* (2020) são SINCERITIES, SINGE e PIDC. Contudo, ao considerar os problemas acurados, o desempenho decresce para a maioria dos algoritmos, apresentando desempenho próximo a preditores aleatórios. Além disso, os algoritmos com melhor desempenho são GENIE3, GRNBOOST2 e PIDC. Exceto pelo PIDC, os algoritmos com melhor desempenho para os dados sintéticos não são os algoritmos com melhor desempenho para dados acurados. Por fim, para dados experimentais, PRATAPA *et al.* (2020) selecionam alguns destes algoritmos com melhor desempenho, considerando também a estabilidade sob diversas execuções e o tempo necessário para o procedimento de inferência. O conjunto final de algoritmos é composto por PIDC, GENIE3, GRNBOOST2, SCODE, PPCOR e SINCERITIES. As conclusões são que GENIE3, PIDC e GRNBOOST2 foram os algoritmos com melhor desempenho para os dados experimentais.

Com essa discussão, queremos ressaltar que não existe garantia que os algoritmos com bom desempenho em problemas sintéticos ou acurados também apresentem bons resultados quando considerados os problemas experimentais. Isso pode ser um indicativo de que a forma de gerar dados sintéticos não esteja sendo capaz de modelar adequadamente as características observadas nos dados experimentais.

Conforme apresentado no início desta seção, avaliar a qualidade de uma rede inferida também permite atribuir qualidade ao algoritmo de inferência. Nesse sentido, algoritmos que geram redes que reproduzem corretamente o fenômeno biológico são ditos melhores que aqueles que não reproduzem corretamente. Além disso, quanto mais próxima for a rede inferida do fenômeno biológico modelado, mais informativa tende a ser essa rede. Dessa forma, tais redes fornecem informações que podem auxiliar no entendimento de diversos processos biológicos.

Diversos aspectos influenciam a avaliação de desempenho do procedimento de inferência e, consequentemente, das GRNs inferidas. Aqui, discutimos (i) a falta de

[11] `https://murali-group.github.io/Beeline/BoolODE.html`

conhecimento sobre a rede *ground-truth* e o uso de diferentes redes de referência e (ii) as métricas usadas para a avaliação.

A falta de conhecimento sobre a rede *ground-truth* e o uso de diferentes redes de referência fazem com que as avaliações das redes inferidas variem substancialmente. As métricas utilizadas para avaliar o desempenho podem enviesar a avaliação da qualidade de uma rede inferida. Além disso, os dados são tipicamente desbalanceados. Por esse motivo, utilizar as métricas apropriadas para essa classe de problemas deve ser considerada.

Como discutido anteriormente, para os problemas *benchmark* propostos por PRATAPA *et al.* (2020), redes *ground-truth* estão disponíveis para os problemas sintéticos e acurados. Já para problemas experimentais, a ausência de tal informação é um problema para a avaliação das redes. Por esse motivo, é comum utilizar três redes de referência (CHEN; MAR, 2018; PRATAPA *et al.*, 2020): STRING, *NonSpecific* e *Cell-type-specific ChIP-Seq*. Essas redes foram apresentadas e explicadas na Seção 2.12.

Um fator importante é o conjunto de métricas utilizados para avaliar as redes inferidas. Na Seção 2.12 foram apresentadas as principais métricas utilizadas para a avaliação de classificadores, em especial, os binários.

No contexto de biologia sistêmica, podemos tomar como base os trabalhos que comparam o desempenho de diferentes algoritmos quando considerando dados scRNA-Seq (PRATAPA *et al.*, 2020; CHEN; MAR, 2018). CHEN; MAR (2018) utiliza como métricas de comparação as áreas sob as curvas ROC e PRC, enquanto PRATAPA *et al.* (2020), introduz, além dessas duas, o conceito de *early precision* (EP) e *early precision ratio* (EPR).

Contudo, não existe padronização em relação ao uso dessas métricas na literatura. Mesmo antes dos trabalhos que mostram o baixo desempenho dos algoritmos de inferência quando considerando dados scRNA-Seq (CHEN; MAR, 2018; PRATAPA *et al.*, 2020), é possível encontrar na literatura discussões sobre o conjunto de métricas mais apropriados para avaliar classificadores binários em dados biológicos, especialmente em conjuntos de dados desbalanceados, como em GRNs.

O estudo desenvolvido por SAITO; REHMSMEIER (2015) fornece uma comparação entre o uso de PRC e ROC na avaliação de classificadores binários quando usando dados biológicos tais como estudos de miRNA e pre-miRNA, preditor de interação regulatória, integração de dados genômicos estruturais e funcionais com redes de proteínas, entre outros. Os autores mostram que, diferentemente das curvas ROC, as curvas PRC expressam a suscetibilidade dos classificadores a conjuntos de dados desequilibrados com pistas visuais claras e permite uma interpretação precisa e intuitiva dos classificadores práticos. Como resultado, os gráficos PRC são recomendados como a ferramenta de análise visual mais informativa.

Já em (CHAN; STUMPF; BABTIE, 2017), onde o algoritmo PIDC é proposto, os

autores ressaltam que os experimentos de *single-cell* para expressão gênica apresentam novos desafios para o preprocessamento de dados. Para avaliar a qualidade das redes inferidas, os autores consideram *precision*, *recall* e a área sob as curvas ROC e PRC, mesmo que o estudo apresentado por SAITO; REHMSMEIER (2015) já houvesse apresentado que ROC não era indicada para dados desbalanceados. Os autores concluem que os algoritmos apresentaram desempenho próximo a preditores aleatórios quando inferindo GRNs a partir de dados scRNA-Seq.

No ano seguinte, BYRON; WANG (2018) apresentam uma revisão comparativa de ferramentas para a inferência de GRNs. Além de apresentar diversos métodos desenvolvidos para a inferência de GRNs, a comparação de desempenho dos algoritmos é realizada levando em consideração as medidas básicas da matriz de confusão (TP, FP, FN, e TN), as métricas básicas *accuracy*, *balanced accuracy*, *precision* e *recall*, além da área sob a curva ROC. Os autores concluem que os resultados obtidos para essas métricas padrão de desempenho são geralmente baixas, para todas as sete ferramentas consideradas, sugerindo que esforços futuros são necessários para o desenvolvimento de ferramentas de inferência de redes mais confiáveis.

Em DELGADO; GÓMEZ-VELA (2019), uma revisão de métodos computacionais para análise e reconstrução de GRNs é apresentada. Em seção específica para discutir a avaliação quantitativa do desempenho dos algoritmos de inferência, os autores enfatizam que o método tradicional de avaliação é comparar a rede inferida com uma rede *gold standard*, permitindo a estimativa de diversas métricas que podem, juntas, fornecer uma avaliação da qualidade do modelo. De acordo com SCHRYNEMACKERS; KÜFFNER; GEURTS (2013), quando a rede *gold standard* é conhecida, as redes inferidas são comparadas utilizando diversas métricas, tais como TPR, TNR, FPR, FNR, *precision*, *F-Score* e as áreas sob as curvas ROC e PRC.

Tanto as curvas ROC e PRC são comumente usadas para comparar o desempenho de diferentes algoritmos na literatura (SMET; MARCHAL, 2010; PRILL *et al.*, 2010; HASE *et al.*, 2013). Contudo, tais métricas foram usadas na literatura principalmente para validar redes baseadas em dados sintéticos (DELGADO; GÓMEZ-VELA, 2019).

ZHAO *et al.* (2021) apresentam uma avaliação crítica das tecnologias de inferência de GRNs. Nesse caso, métodos clássicos para inferência de GRNs são discutidos e utilizados para inferir GRNs utilizando tanto dados simulados quando reais. A avaliação é realizada considerando as quatro medidas básicas da matriz de confusão, além de FPR, TPR, *precision* e *recall*. Além disso, foram utilizadas as AUROC e AUPRC. Os autores concluem que cada método possui um campo de aplicação mais adequado. Por exemplo, métodos *model-based* são preferíveis para inferir redes pequenas utilizando dados reais. Já os métodos baseados em teoria da informação são melhores para resolver problemas que utilizam somente dados na forma *steady-state*. Métodos baseados em aprendizado tendem

a apresentar melhores resultados na inferência de GRNs grandes. Contudo, os autores ressaltam que os resultados em dados reais são geralmente inferiores daqueles obtidos para dados simulados, indicando, novamente, que ainda é desafiador inferir GRNs de organismos reais.

Em 2020, uma discussão cujo foco concentra-se na inexistência de um consenso sobre o conjunto de métricas a serem utilizados para avaliar classificadores binários no contexto de conjuntos de dados biomédicos e de bioinformática é apresentada (CHICCO; JURMAN, 2020). Os autores discutem que *accuracy* e *F1-Score* estão dentre as métricas mais popularmente adotadas, mas que elas podem apresentar resultados otimistas, especialmente em conjunto de dados desbalanceados, enviesando a análise dos resultados obtidos pelos modelos. Usando esse argumento, os autores mostram que o *Matthews Correlation Coefficient* (MCC) é uma métrica estatística mais confiável, uma vez que valores maiores só são produzidos quando a predição obtém bons resultados em todas as quatro métricas básicas da matriz de confusão em proporção ao tamanho de seus elementos. MCC é calculado como $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$. Os autores concluem, através de uma explicação inicial sobre as propriedades matemáticas e a avaliação do MCC em seis casos sintéticos e em um cenário genômico real, que MCC produz um *score* mais informativo e confiável na avaliação de classificadores binários do que a *accuracy* e o *F1-Score*. Por esse motivo, os autores sugerem que o MCC deve ser preferido.

Baseado em estudos no cenário de bioinformática, CHICCO; TÖTSCH; JURMAN (2021) afirmam que MCC é mais confiável que a *balanced accuracy*, *bookmaker informedness*, e *markedness* na avaliação da matriz de confusão para problemas de duas classes. Novamente, sugere-se que MCC seja adotado como métrica padrão. Contudo, os autores enfatizam que a comunidade científica não tem um acordo comum sobre indicadores estatísticos de uso geral para avaliar matrizes de confusão de duas classes. Uma discussão similar, agora considerando o *Cohen's Kappa* e o *Brier Score* é apresentado em (CHICCO; WARRENS; JURMAN, 2021). As propriedades matemáticas e as relações entre MCC, *Cohen's Kappa* e *Brier Score* são explicadas. Os autores mostram que cada métrica gera diferentes valores, levando a resultados discordantes. A partir disso, os autores concluem que MCC fornece um resultado mais verdadeiro e informativo, aconselhando novamente o uso do MCC.

Por fim, os mesmos autores sugerem em (CHICCO; JURMAN, 2023) que MCC deve substituir AUROC como a métrica padrão para avaliar classificadores binários. Diferentemente da discussão inicial apresentada em (SAITO; REHMSMEIER, 2015), os autores mostram que valores altos de MCC (em torno de 0.9) sempre correspondem a altos valores de AUROC. Entretanto, a recíproca não é verdadeira.

Contudo, comparar diferentes *datasets* e o desempenho de diferentes algoritmos em conjunto, tende a ser ainda mais complicado. Utilizar o valor absoluto de uma única

métrica, tal como ACC ou MCC pode não ser suficiente informativo quando considera-se um conjunto amplo de *datasets*. Nesse caso, uma alternativa são os PPs (DOLAN; MORÉ, 2002; BARBOSA; BERNARDINO; BARRETO, 2010), conforme usado em diversos trabalhos e apresentados na Seção 2.12 (SILVA *et al.*, 2021, 2023, 2024).

A partir dos PPs, é possível extrair (DOLAN; MORÉ, 2002): (i) a abordagem com maior $\rho(1)$ é aquela com os melhores resultados para a maioria dos problemas em P, (ii) a abordagem mais confiável é aquela com o menor τ tal que $\rho(\tau) = 1$, e (iii) o melhor desempenho geral é obtido pela abordagem com a maior área sob a curva do PP.

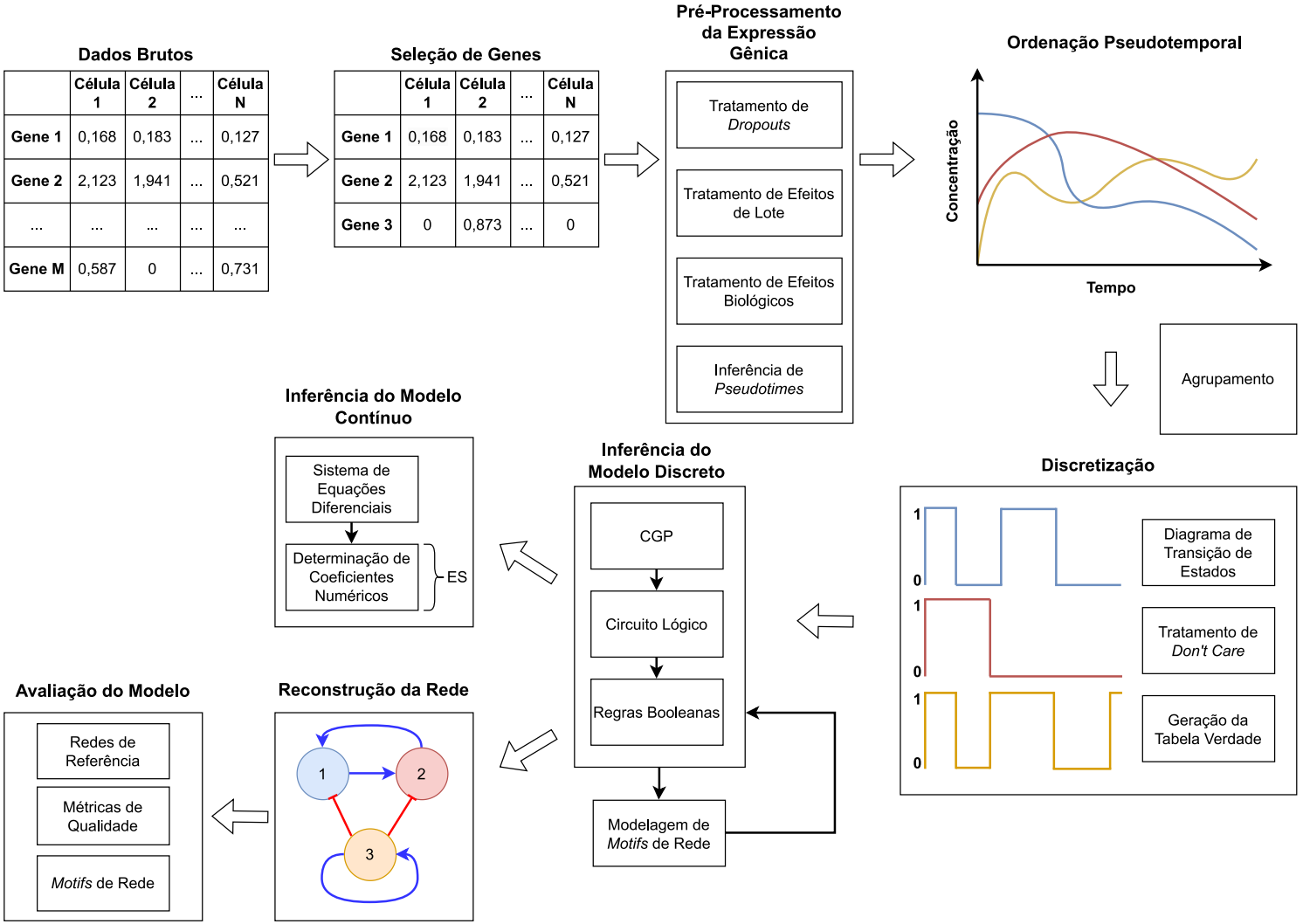
Como discutido, diversas métricas têm sido utilizadas para avaliar as redes inferidas e comparar os algoritmos de inferência. Ainda que estudos tenham mostrado que a ROC não é adequada para dados desbalanceados, tal curva ainda é utilizada para comparações (CHEN; MAR, 2018; PRATAPA *et al.*, 2020). Além disso, no contexto de classificadores binários, as métricas básicas tais como *accuracy*, *precision* e *recall*, além das métricas de primeiro nível, tais como *balanced accuracy*, *informedness*, *F-Measure* e *Cohen's Kappa* geram valores diferentes, levando a resultados discordantes. Ainda, quando deseja-se comparar um único *dataset* inferido por diferentes algoritmos, o uso de um valor absoluto de uma métrica ou uma curva pode ser uma boa alternativa. Contudo, comparar conjuntos diferentes de *datasets*, coim diferentes características, e considerando diferentes algoritmos, o uso das métricas básicas podem não fornecer informação suficiente para concluir que um algoritmo é superior a outro (SILVA *et al.*, 2023).

Dados tais problemas, uma avaliação sistemática dos valores obtidos por tais métricas e a informação que pode ser obtida com a interpretação de tais valores torna-se necessária.

3 MÉTODO PROPOSTO

Neste capítulo é apresentado o método proposto para a inferência e avaliação de GRNs, denominado CGPGRN. Dois problemas de otimização são abordados neste trabalho. Um consiste em encontrar um circuito lógico combinacional, com o menor número de elementos lógicos possíveis, que descrevam as relações regulatórias entre os genes, a partir de dados de expressão gênica. Já o segundo, consiste em determinar os coeficientes numéricos de um sistema de equações diferenciais ordinárias. Enquanto o primeiro objetiva determinar o menor circuito capaz de implementar as regras lógicas obtidas no processo de discretização dos dados de expressão gênica, fornecendo um modelo discreto de GRN, qualitativo, o segundo determina os parâmetros cinéticos do modelo contínuo equivalente ao discreto, fornecendo um modelo contínuo de GRN, quantitativo. Os detalhes desses problemas são descritos em seções posteriores. O processo do CGPGRN é apresentado no fluxograma da Figura 27.

Figura 27 – Fluxograma do procedimento desenvolvido.



Fonte: Desenvolvido pelo autor (2023).

Um conjunto de dados brutos é utilizado e os genes de interesse são selecionados. Qualquer tipo de dado, oriundo de qualquer tecnologia de perfilamento transcricional pode ser utilizado. O único requisito é que exista a informação temporal associada à cada nível de concentração de expressão gênica. A seleção de genes de interesse pode ser realizada por algoritmos de seleção de características ou informação resultante de experimentos biológicos. O CGPGRN recebe como entrada um conjunto de dados de expressão gênica e realiza um pré-processamento, contendo quatro etapas principais: (i) tratamento de *dropouts*, (ii) tratamento de efeitos de lote, (iii) tratamento de efeitos biológicos e (iv) inferência de pseudotimes, se necessário. Após isso, os dados são ordenados de acordo com a informação pseudotemporal. Isso coloca os dados na forma de séries temporais. Em seguida, a etapa de agrupamento é opcional, e pode ser utilizada como seleção de subconjuntos de genes e direcionamento do procedimento de busca. Os dados, já na forma de séries temporais, são discretizados. Neste processo, são obtidos diagramas de transição de estados, as situações de irrelevância são tratadas e obtém-se uma tabela verdade. Essa tabela verdade é fornecida como entrada para o algoritmo de busca (CGP), responsável por obter um circuito lógico que descreva os estados observados na etapa de discretização. As expressões lógicas (regras Booleanas) são extraídas do circuito lógico evoluído. Durante o processo de inferência, é possível selecionar a modelagem de *motifs* de rede, em específico, permitir, ou não, a autorregulação. Uma vez que o modelo Booleano é obtido, este pode ser usado tanto para (i) a inferência do modelo contínuo, quanto para (ii) reconstrução da rede. Enquanto (i) fornece um modelo na forma de um sistema de equações diferenciais, cujos coeficientes numéricos são determinados através de ES, (ii) gera uma rede no formato de probabilidade de ocorrência das regulações, com o tipo de regulação. Tais informações são utilizadas para a avaliação do modelo, onde considera-se as redes de referência, as métricas de qualidade e a seleção de *motifs* de rede.

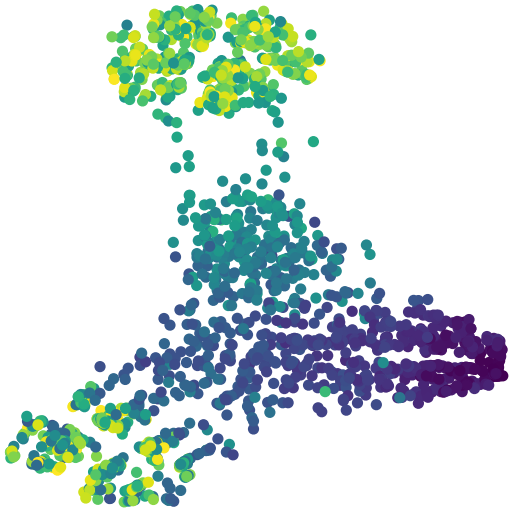
Cada etapa é discutida nas seções que seguem.

3.1 PRÉ-PROCESSAMENTO E ORDENAÇÃO PSEUDOTEMPORAL

Neste trabalho são utilizados três tipos de dados: (i) simulados, (ii) scRNA-Seq, e (iii) dados de organismos amplamente estudados na literatura. Os dados simulados consistem das medições de níveis de expressão gênica ao longo do tempo de modelos de EDOs publicados na literatura. Já os dados de scRNA-Seq são divididos em três grupos: (i) sintéticos, (ii) acurados e (iii) experimentais. Segundo PRATAPA *et al.* (2020) os dados síntéticos e acurados(*curated*) são capazes de representar com fidelidade as características presentes em dados extraídos utilizando as técnicas de perfilamento por scRNA-Seq. Tais dados incluem *dropouts* e informações sobre o *pseudotime*. É importante destacar que, muitas vezes, as células podem ter diferentes *steady-states* durante seu desenvolvimento biológico. Isso é visível na diferenciação celular, por exemplo. Além disso,

células de mesmo tipo podem apresentar concentrações proteicas diferentes, conhecido como heterogeneidade celular. Os diferentes *steady-states* são facilmente identificados utilizando-se dos *pseudotimes*, como apresentado na Figura 28. Essa figura foi gerada utilizando o t-SNE, que consiste de um método estatístico para visualizar dados de alta dimensão. Aqui considera-se o uso do *Slingshot* (STREET *et al.*, 2018) pois é apontado como o melhor método para a inferência de *pseudotimes* e utilizado em (PRATAPA *et al.*, 2020). Por fim, os dados de organismos amplamente estudados na literatura incluem subredes de *E. coli* e *S. cerevisiae* com a adição, ou não, de ruídos.

Figura 28 – Visualização por t-SNE de dados de expressão gênica perfilados por scRNA-Seq. Cada ponto representa uma célula que é colorida em relação ao seu ponto no tempo. Quanto mais escura é a cor, menor é o ponto no tempo.

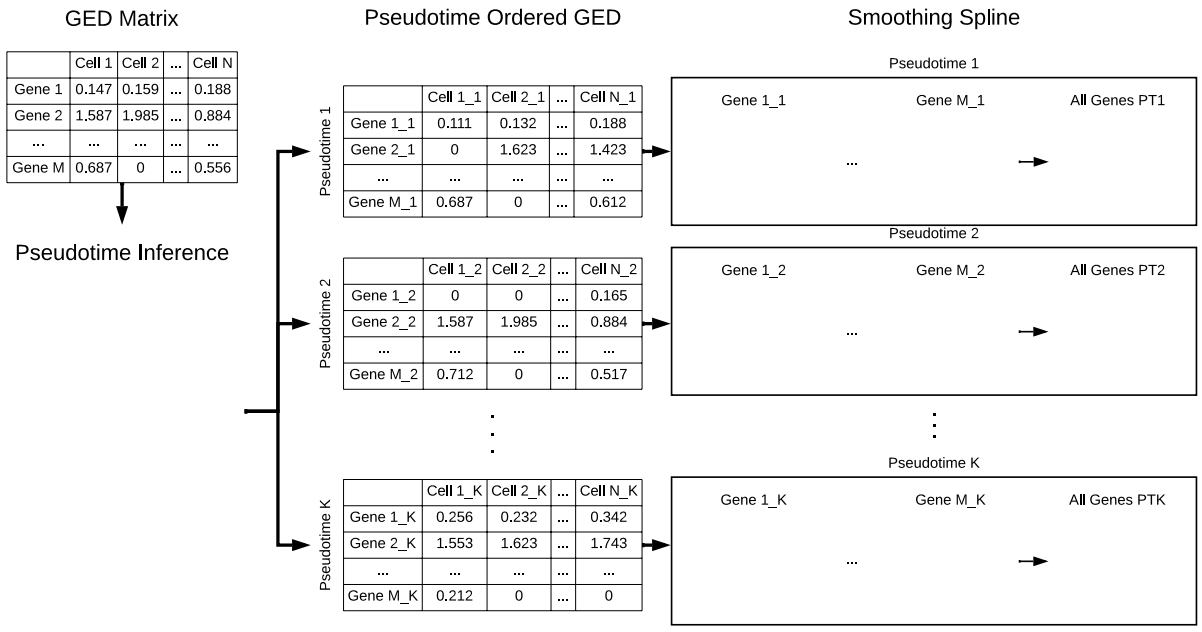


Fonte: Elaborado pelo autor (2022).

Os dados obtidos através de simulação são considerados já tratados, pois possuem os pontos do tempo determinados com seus respectivos valores de expressão gênica. Entretanto, para dados scRNA-Seq, justamente pelas características de *dropout* e falta da informação de tempo físico, faz-se necessário um pré-processamento. O fluxograma geral deste pré-processamento é apresentado na Figura 29 e discutido em detalhes a seguir.

A Figura 30 ilustra os dados brutos, onde cada ponto representa uma célula. Consideramos aqui uma matriz A' de dados de expressão gênica M $\times$ N, onde M são os genes e N são as células. O primeiro passo para o pré-processamento é ordenar as células de acordo com seu *pseudotime*. Tal informação temporal deve ser fornecida como entrada para o *framework*. A inferência dessa informação para a tecnologia de perfilamento por *scRNA-Seq* foi apresentada na Seção **2.2.5**. Caso exista mais de um *pseudotime*, o pré-processamento é aplicado em cada trajetória de *pseudotime*, isto é, obtém-se K novas matrizes $A'_1, \ldots A'_K$ onde, para cada *pseudotime* existe uma distribuição dos pontos de

Figura 29 – Fluxograma de pré-processamento. Dada uma matriz de dados de expressão gênica (*GED Matrix*), o *pseudotime* inferido (*Pseudotime Inference*) é utilizado para gerar K novas matrizes, uma para cada *pseudotime*. Essas matrizes têm suas células ordenadas de acordo com a informação do *pseudotime* (*Pseudotime Ordered GED*) e, para cada matriz e para cada gene, os dados são aproximados usando *cubic smoothing spline* a fim de remover ou suavizar os efeitos de variações técnicas e biológicas.

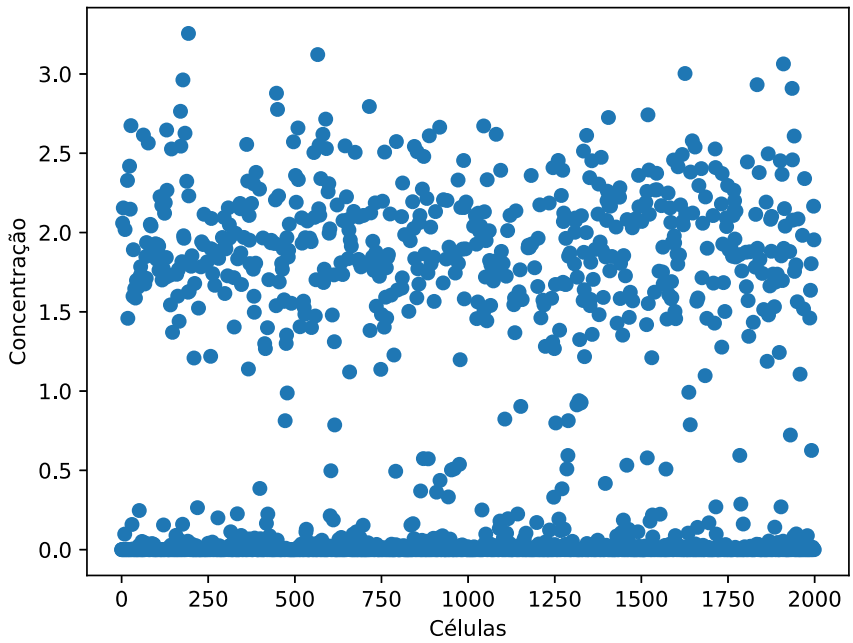


Fonte: SILVA *et al.* (2023).

expressão gênica sobre o tempo. Essa simples ordenação já facilita a visualização dos dados, conforme apresentado na Figura 31.

Apesar do *Slingshot* assumir que as células estão arranjadas em um *pseudotime* que varia entre 0 e 1, o intervalo adotado no *pseudotime* não interfere no pré-processamento dos dados. Qualquer intervalo pode ser usado, contanto que o resultado final seja a distribuição dos pontos de expressão gênica ao longo do tempo. Como discutido na Seção **2.2.5**, diversos fatores podem afetar a confiabilidade e veracidade da informação que pode ser obtida utilizando dados resultantes de perfilamento scRNA-Seq. Alguns desses fatores são obtidos graças às variações técnicas tais como *batch effects*, eficiência de captura específica de célula, viés de amplificação e *dropouts* (HWANG; LEE; BANG, 2018). *Dropouts* podem ocorrer pois a maioria dos genes é utilizada em apenas um subconjunto de tipos celulares, e outros genes não são detectados, ainda que estejam expressados. Isso resulta em uma matriz de expressão gênica com muitos zeros, e é um grande desafio para a análise computacional (ANDREWS *et al.*, 2020), uma vez que é difícil distinguir e modelar apropriadamente as fontes dos zeros observados. Além disso, a variação biológica pode introduzir erros, tais como a natureza estocástica da expressão gênica e os efeitos criados pelo ciclo celular (HWANG; LEE; BANG, 2018). Desta forma, o passo seguinte

Figura 30 – Dados brutos resultantes de um perfilamento de scRNA-Seq com 2.000 células. É possível perceber a quantidade de *dropouts* (pontos onde o eixo das oordenadas é zero).



Fonte: Elaborado pelo autor (2022).

consiste na remoção dos *dropouts*. Discussões mais detalhadas sobre pré-processamento, de maneira geral, foram apresentadas na Seção 2.13. Contudo, esse passo é fundamental pois é responsável por não enviesar o resultado dos *cubic smoothing splines* que serão aplicados no passo seguinte. Após a remoção dos *dropouts*, os dados ficam conforme apresentado na Figura 32.

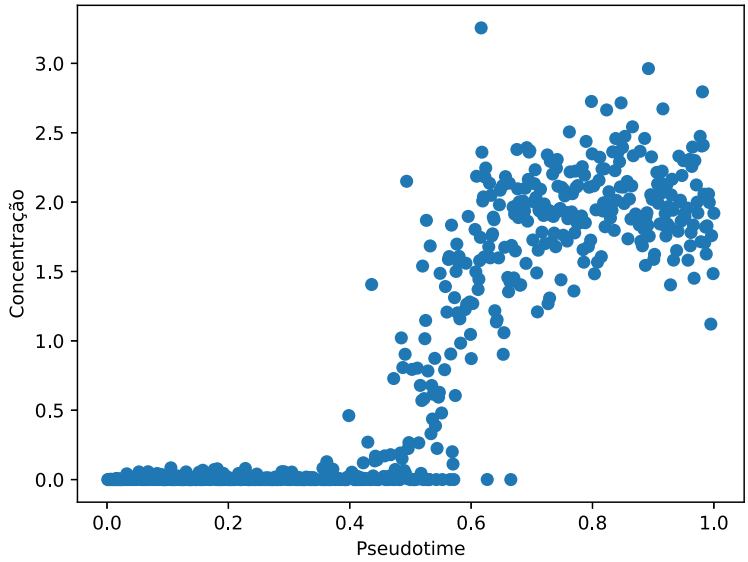
O último passo do pré-processamento consiste em aproximar os dados de expressão gênica a fim de reduzir ou aliviar os efeitos de variações técnicas e biológicas, bem como obter uma curva única que represente a expressão gênica ao longo do tempo, através de *cubic smoothing splines*. O *smoothing spline* f minimiza

$$b\sum_{j=1}^{m}\omega_j \mid y_i - f(x_j) \mid^2 + (1-b)\int \lambda(t) \mid D^2 f(t) \mid^2 \mathrm{d}t \tag{3.1}$$

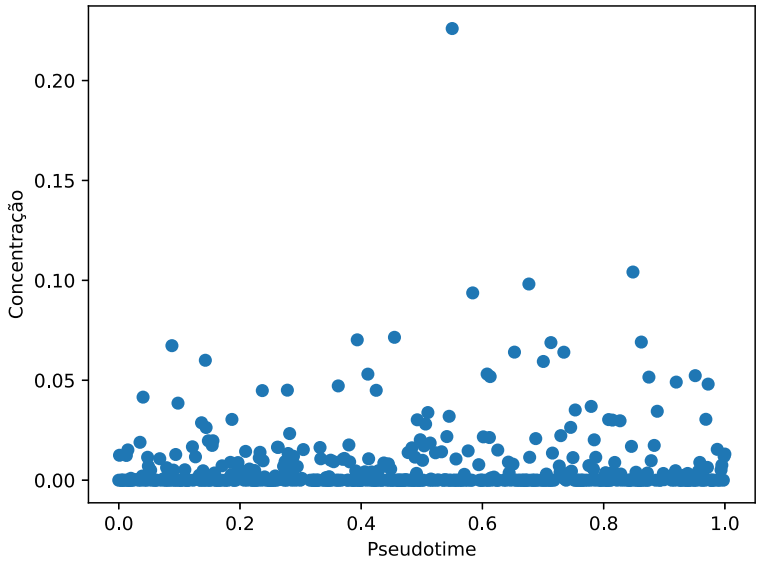
onde o primeiro termo é a medida de erro e o segundo é a medida de severidade. Ainda, m é o número de dados de entrada. x_j e y_j referem-se à j-ésima entrada de x (pontos de *spline*) e y, respectivamente. $D^2 f$ denota a segunda derivada da função f. ω_j é o peso de medida do erro e, por padrão, é 1. O valor padrão para a função de peso constante por partes λ na medida de severidade é a função constante 1. Além disso, b é o valor de suavização, limitado por 0, o que significa que o *smoothing spline* é a reta de mínimos quadrados ajustada aos dados, e 1, que é o interpolador de *spline* cúbico natural. Como resultado, obtém-se as curvas apresentadas na Figura 33.

Figura 31 – Dados de expressão gênica separados por *pseudotime*.

(a) Células que compõem o primeiro *pseudotime*.



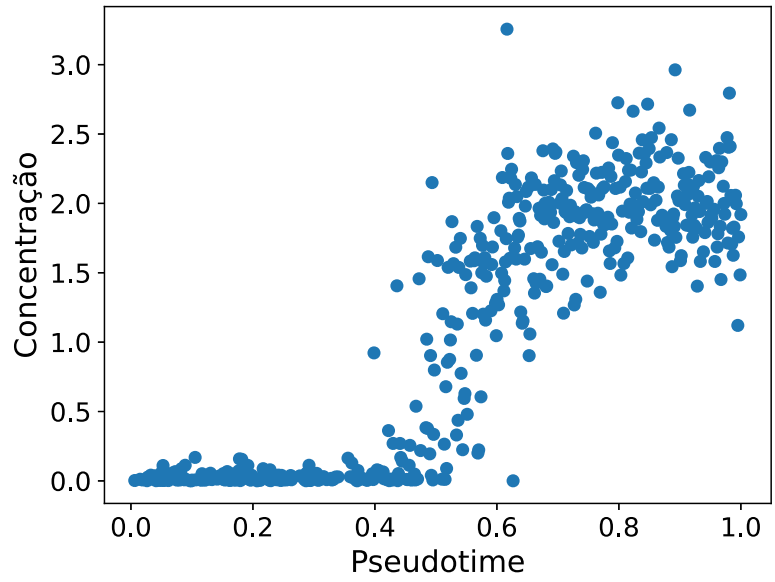
(b) Células que compõem o segundo *pseudotime*



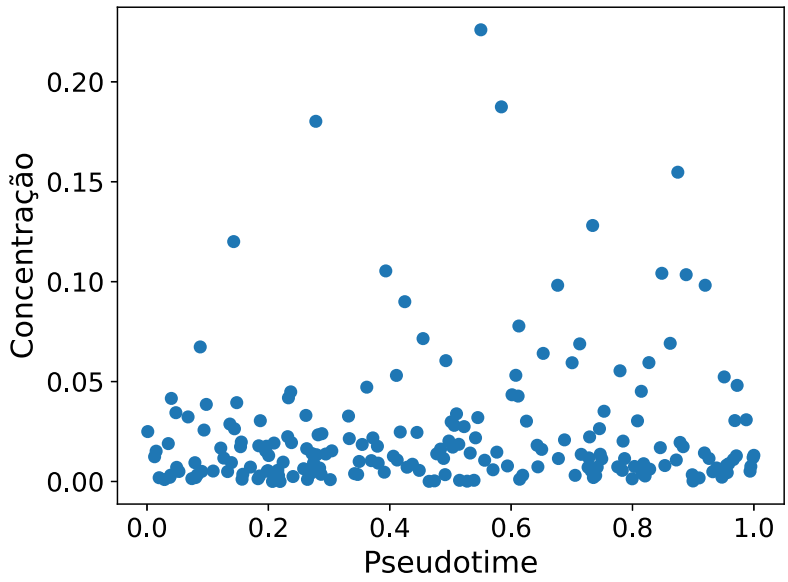
Fonte: Elaborado pelo autor (2022).

Figura 32 – Dados de expressão gênica separados por *pseudotime* e com remoção de *dropouts*.

(a) Células que compõem o primeiro *pseudotime*.



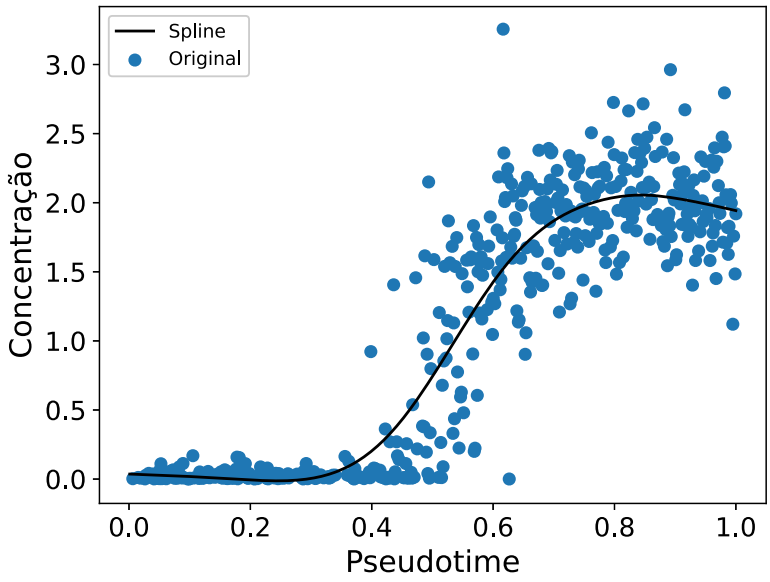
(b) Células que compõem o segundo *pseudotime*



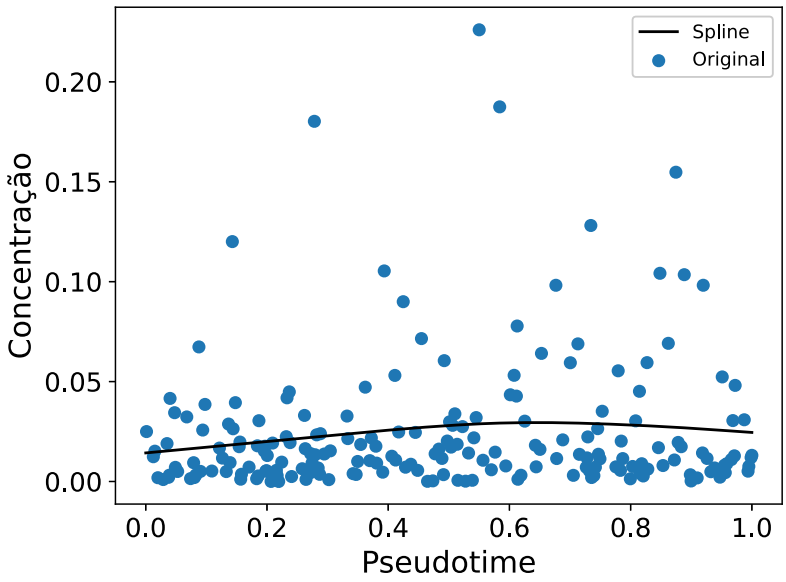
Fonte: Elaborado pelo autor (2022).

Figura 33 – Dados de expressão gênica separados por *pseudotime*, com remoção de *dropouts* e com suavização via *smoothing splines*.

(a) Células que compõem o primeiro *pseudotime*.



(b) Células que compõem o segundo *pseudotime*



Fonte: Elaborado pelo autor (2022).

3.2 AGRUPAMENTO

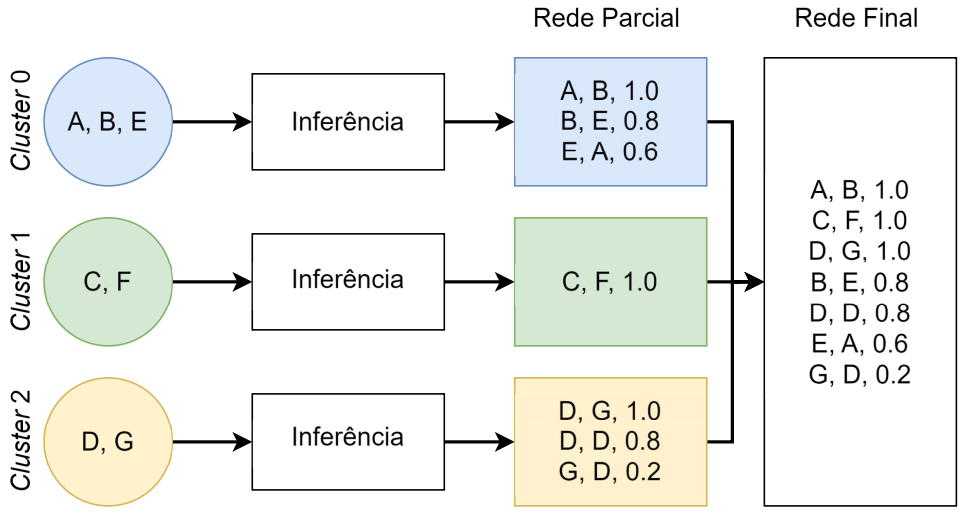
Conforme ressaltado anteriormente, nem todos os genes estão expressos em todas as células o tempo todo. Por este motivo, a literatura utiliza técnicas para a seleção de genes mais variantes e tenta aplicar critérios para separar os genes em grupos nos quais eles estejam correlacionados (ALANNI *et al.*, 2019; SILVER *et al.*, 2006; LIANG *et al.*, 2020).

Isso se torna ainda mais relevante quando lida-se com dados experimentais. As tecnologias de perfilamento (ver Seção **2.2.5**), em especial a scRNA-Seq, gera uma grande quantidade de medidas considerando milhares de células com milhares de genes.

A etapa de agrupamento é um passo não obrigatório para o usuário, e é realizada logo após o pré-processamento. Seu objetivo é reduzir o número de possíveis reguladores para cada gene.

Uma abordagem amplamente conhecida para *clustering* é o *k-means*, que separa as amostras em grupos de mesma variância. Esses *clusters* são gerados a partir da minimização de um critério conhecida como inércia ou WCSS, discutido nas Seções **2.11.1** e 3.2. Desta forma, assumimos que cada gene é regulado pelos genes que estão no mesmo *cluster* e, para cada *cluster*, uma rede parcial é obtida. Desta forma, a GRN final é resultante da união das redes parciais, conforme apresentado na Figura 34.

Figura 34 – Unificação das redes parciais para cada *cluster*. Cada *cluster* tem sua inferência realizada de maneira independente. As redes parciais são obtidas com as informações (Regulador, Alvo, Probabilidade). A rede final é construída unificando as redes parciais e ordenando por maior probabilidade.



Fonte: Elaborado pelo autor (2024).

É importante ressaltar que a união das redes parciais não criará uma rede onde os *clusters* compartilham relações regulatórias pois, por definição, os *clusters* são disjuntos. Com isso, por GRN final entende-se, então, a unificação das relações regulatórias obtidas

em cada um dos *clusters* em um arquivo único para posterior avaliação. A unificação das redes parciais é discutida em maiores detalhes na Seção 3.4.

Contudo, as espécies que estão sendo modeladas podem ser agrupadas de diversas maneiras. Conforme discutido na Seção 2.9 podem ser utilizadas técnicas de *clustering* hierárquico ou até mesmo coeficientes de correlação. O *framework* CGPGRN conta com KMeans e *clustering* hierárquico aglomerativo, onde o melhor número de *clusters* é determinado pelo coeficiente de silhueta para o primeiro e por DBI para o segundo, para um dado número máximo de clusters a ser testado, além de agrupamento por coeficientes de correlação de Spearman, Pearson e Kendall Tau, onde o valor de *threshold* é passado como argumento. O impacto do uso de diferentes técnicas de agrupamento é discutido na Seção **4.6.2**.

3.3 DISCRETIZAÇÃO

Uma vez que os dados já passaram pelo pré-processamento, e os dados contínuos já foram obtidos através do *spline*, faz-se necessária uma discretização para que estes dados representem sinais lógicos de tal maneira que possa-se obter um diagrama de transição de estados e consequentemente uma tabela verdade. Essa tabela verdade, por sua vez, é fornecida como entrada para o algoritmo de inferência. Aqui, a CGP é adotada, e é responsável por determinar as relações regulatórias (lógicas) entre os genes. Este processo é ilustrado na Figura 35.

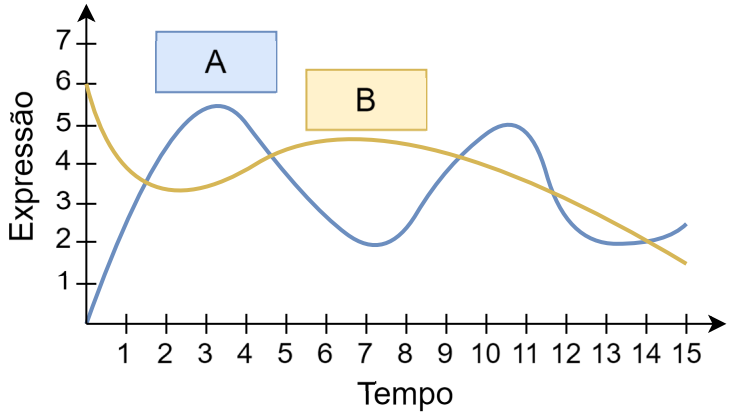
Conforme discutido na Seção **2.11.1**, diversos métodos de discretização podem ser utilizados (GALLO *et al.*, 2015; BECQUET *et al.*, 2002; PENSA *et al.*, 2004). No contexto de inferência de GRNs com dados de scRNA-seq, realizamos um estudo (SILVA *et al.*, 2021) sobre o impacto desses métodos de discretização, que gera perda de informação e consiste em uma etapa crucial para o sucesso do algoritmo de inferência, como será discutido na Seção **4.6.3**.

O *framework* CGPGRN conta com diversos métodos de discretização, a saber: *Mean*, *Median*, TSD, EFD, *KMeans* e BiKMeans. Além disso, desenvolvemos um novo método de discretização (DSSPD) (SILVA *et al.*, 2024), que será discutido em detalhes na Seção **3.3.1**. Existem também métodos *Ensemble* disponíveis no *framework*, que combinam essas discretizações. A discretização também pode ser aplicada separadamente em cada *cluster*, se desejado.

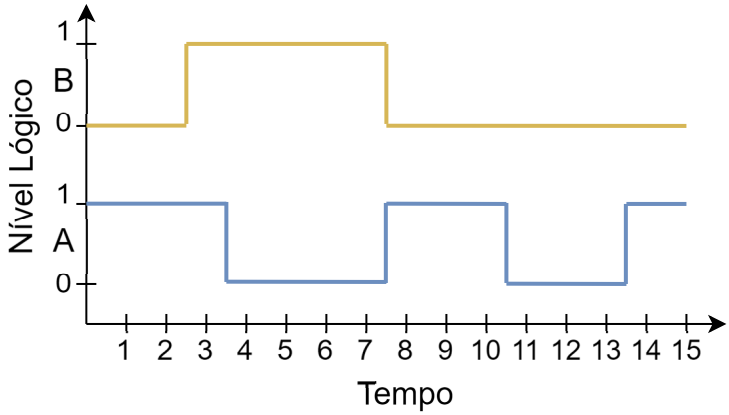
Contudo, independentemente do método de discretização utilizado, durante a determinação do diagrama de transição de estados, pode-se obter transições ambíguas, isto é, um mesmo estado gerando a possibilidade de dois estados distintos. Isso vai contra a definição de circuitos lógicos combinacionais e deve ser tratado. Para ilustrar esta situação, considere os dados discretizados apresentados na Figura 36a e o diagrama de transição de estados obtidos a partir dessa discretização na Figura 36b. Segundo o diagrama de

Figura 35 – Processo de discretização.

(a) Dados resultantes do pré-processamento.



(b) Dados discretizados.



(c) Diagrama de transição de estados

Estados Discretos

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1
B	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

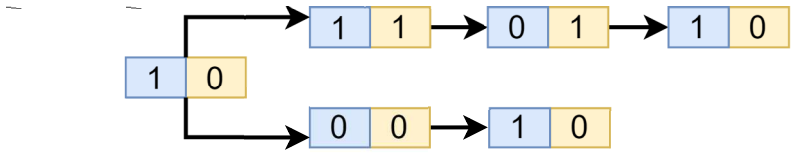


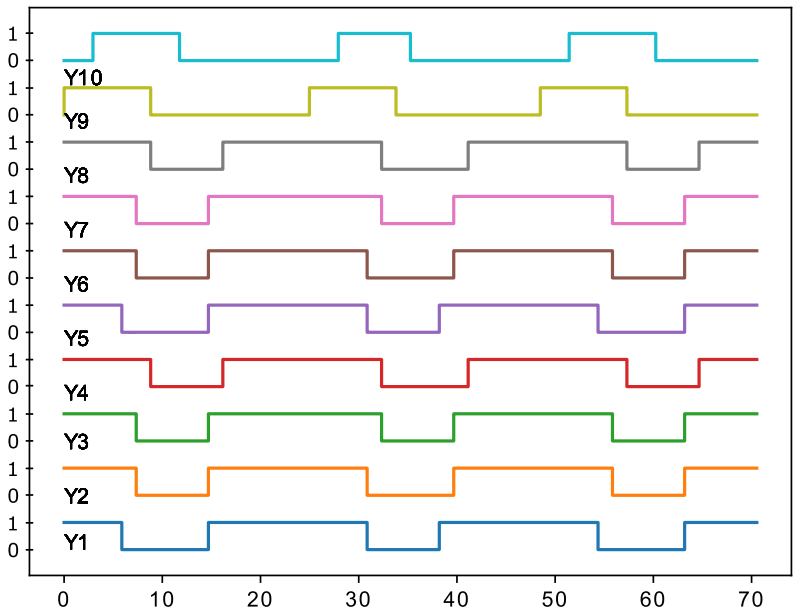
Diagrama de Transição de Estados

Fonte: Elaborado pelo autor (2024).

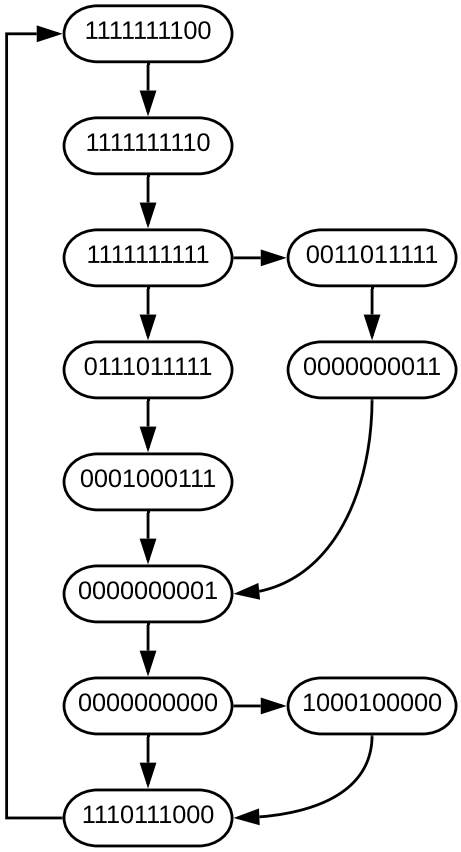
transição de estados, é possível perceber que existem duas ramificações, uma no estado 1111111111 e outra no estado 0000000000. A fim de remover essas ambiguidades, o primeiro passo é analisar, segundo a discretização, a quantidade de vezes em que o sistema discretizado segue por cada ramificação. Essa informação é apresentada na Figura 37.

Figura 36 – Ambiguidade em transições de estados.

(a) Dados discretizados.



(b) Diagrama de transição de estados.

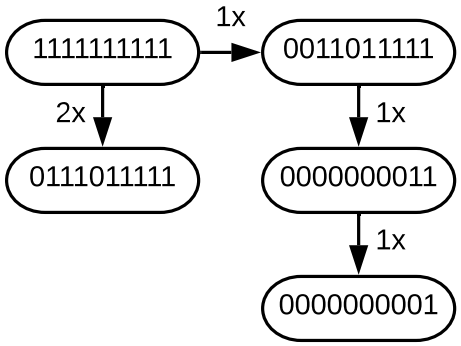


Fonte: SILVA *et al.* (2020).

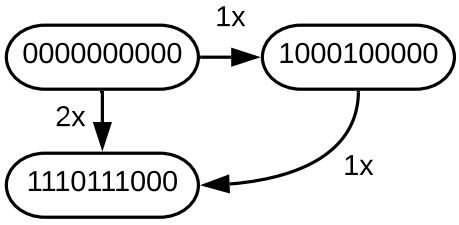
Para a transição da Figura 37a, percebe-se que o sistema discretizado vai duas

Figura 37 – Contagem de transições do sistema discretizado.

(a) Ambiguidades na transição do estado 1111111111.



(b) Ambiguidades na transição do estado 0000000000.



Fonte: SILVA *et al.* (2020)

vezes para o estado 0111011111 e uma única vez para o estado 0011011111, e para a transição da Figura 37b, o sistema vai duas vezes para o estado 1110111000 e uma vez para o estado 1000100000. Entretanto, o estado 1000100000 retorna ao estado da primeira ramificação 1110111000. Uma forma de resolver a ambiguidade é simplesmente optar pela ramificação que ocorre mais frequentemente. Isso resultaria em transição para 0111011111 do estado 1111111111 e 1110111000 do estado 0000000000. Para o caso da Figura 37b, como existe um ciclo, outra forma de resolver seria obrigar o estado 0000000000 ir para 1000100000 e depois para o estado 1110111000. Contudo, uma terceira forma de resolver as ambiguidades é analisar *bit* a *bit* as possíveis transições. Considerando, então, a ramificação da Figura 37a, de maneira similar à primeira forma de solução, pode-se verificar quais são os *bits* que diferem entre as transições e optar por aqueles onde cada *bit* tem maior ocorrência e tratar como irrelevância aqueles onde existe um empate na frequência de ocorrência. Isso é exemplificado na Tabela 5.

Tabela 5 – Terceira forma de resolver as ambiguidades nas transições de estado.

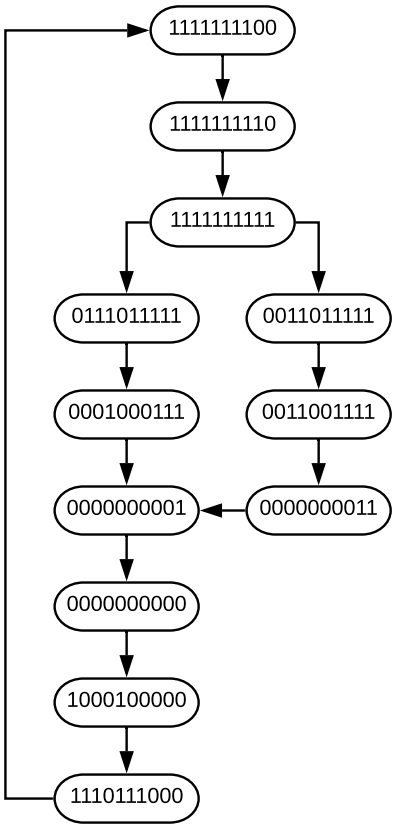
Transição	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	0	1	1	1	0	1	1	1	1	1
2	0	0	1	1	0	1	1	1	1	1
Final	0	“X”	1	1	0	1	1	1	1	1

Fonte: SILVA *et al.* (2020).

Portanto, uma possível solução para as ambiguidades apresentadas na Figura 36b seria utilizar a terceira forma para a transição do estado 1111111111 e utilizar a segunda forma para a transição do estado 0000000000. Isso resultaria em um diagrama de transição de estados final, conforme apresentado na Figura 38.

Como existem diversas maneiras de tratar as ambiguidades apresentadas nos

Figura 38 – Diagrama de transição de estados final para as ambiguidades apresentadas na Figura 36b. Ainda que aparentemente exista uma ramificação, a solução apresentada na Tabela 5 introduz um estado de irrelevância. Desta maneira, o algoritmo de inferência será responsável por determinar qual das ramificações as relações regulatórias serão obtidas.



Fonte: SILVA *et al.* (2020).

diagramas de transição de estados, neste trabalho optaremos por utilizar a terceira solução, analisando sempre *bit* a *bit* em cada uma das ambiguidades encontradas no sistema discretizado pois, além de ser uma solução simples de implementar, dá-se preferência para o estado que ocorre com maior frequência e, em caso de empate, o estado de irrelevância é resolvido pelo algoritmo de inferência.

Outra questão importante é a montagem da tabela verdade final que será dada como entrada para o algoritmo de inferência. Uma vez definido o diagrama de transição de estados final, basta montar a tabela respeitando-se as transições do diagrama. Por conseguinte, a tabela sempre terá n_g entradas e saídas, onde n_g é o número de genes presentes nos dados. Além disso, é possível perceber que os diagramas de transição de estado não apresentaram todos os estados possíveis. No exemplo apresentado com 10 genes, existiriam $2^{10} = 1024$ combinações das entradas. Entretanto, somente 12 estados foram identificados no processo de discretização. Isso pode acontecer pois os dados correspondem a uma observação curta do fenômeno que está sendo modelado ou algumas transições de fato nunca ocorrem (SILVA *et al.*, 2020). Desta forma, todos os estados não obtidos no sistema discretizado serão tratados como irrelevantes. A Tabela 6 mostra como ficaria

Tabela 6 – Tabela verdade final resultante do diagrama de transição de estados da Figura 38.

I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	X	X	X	X	X	X	X	X	X	X
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	1	1	1	0	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	0	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0	X	1	1	0	1	1	1	1	1

Fonte: Elaborado pelo autor (2022).

a tabela verdade obtida a partir do diagrama de transição de estados da Figura 38, considerando somente os estados obtidos no processo de discretização e um exemplo de um estado não presente no sistema discretizado (0000000010).

3.3.1 *Distribution and Successive Spline Points Discretization*

É comum encontrar na literatura métodos, para discretização ou não, que assumem que os dados de expressão gênica possuem distribuições normais. Contudo, essa afirmação é incorreta na maioria das vezes (TORRENTÉ *et al.*, 2020; MARKO; WEIL, 2012).

Tendo em vista as características dos dados oriundos de perfilamento por scRNA-Seq, propomos um método de discretização que leva em consideração a distribuição dos dados denominado Discretização Baseada na Distribuição de Dados e Pontos Sucessivos de *Spline* (DSSPD - do inglês *Distribution and Successive Spline Points Discretization*). Contudo, ao invés de utilizar os dados de maneira bruta, isto é, contendo todos os efeitos de variação biológica, efeito de lote e *dropouts*, o DSSPD utiliza o pré-processamento apresentado na Seção 3.1 antes de iniciar seu procedimento.

Com os dados pré-processados, o primeiro passo do DSSPD consiste em identificar

a distribuição dos dados de expressão gênica para cada gene. Aqui estamos interessados em dividir os genes em dois grupos:

1. aqueles cuja expressão gênica é majoritariamente baixa, e
2. aqueles cujo nível de expressão gênica apresenta variabilidade ao longo do tempo

Por esse motivo, consideramos somente dois tipos de distribuição: exponencial e normal, para os casos (1) e (2), respectivamente.

A distribuição de cada conjunto de dados de expressão gênica de cada gene é analisada para verificar se elas são oriundas de uma distribuição normal ou exponencial. Para determinar em qual distribuição os dados de expressão gênica melhor se encaixam, utilizamos o teste de Kolmogorov-Smirnov (BERGER; ZHOU, 2014).

O teste de Kolmogorov-Smirnov (teste KS) é usado para comparar duas distribuições a fim de determinar se elas estão oriundas de uma mesma distribuição subjacente. Além disso, é um teste não paramétrico.

O teste KS fornece um valor numérico relacionado à diferença das distribuições de dois conjuntos de dados. Desta forma, o teste KS pode ser definido como o valor máximo da diferença entre as funções de distribuição acumulada (CDF - do inglês *cumulative distribution function*) dos dois conjuntos de dados.

Um CDF empírico é criado ordenando os dados por valor e criando um histograma, dado por:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{[-\infty,x]}(X_i) \tag{3.2}$$

em que $I_{[-\infty,x]}(X_i)$ é a função indicadora, igual a 1 se $X_i \leq x$ e igual a 0, caso contrário.

A estatística de Kolmogorov-Smirnov para uma dada função de distribuição acumulada $F(x)$ é:

$$D_n = \sup_x |F_n(x) - F(x)| \tag{3.3}$$

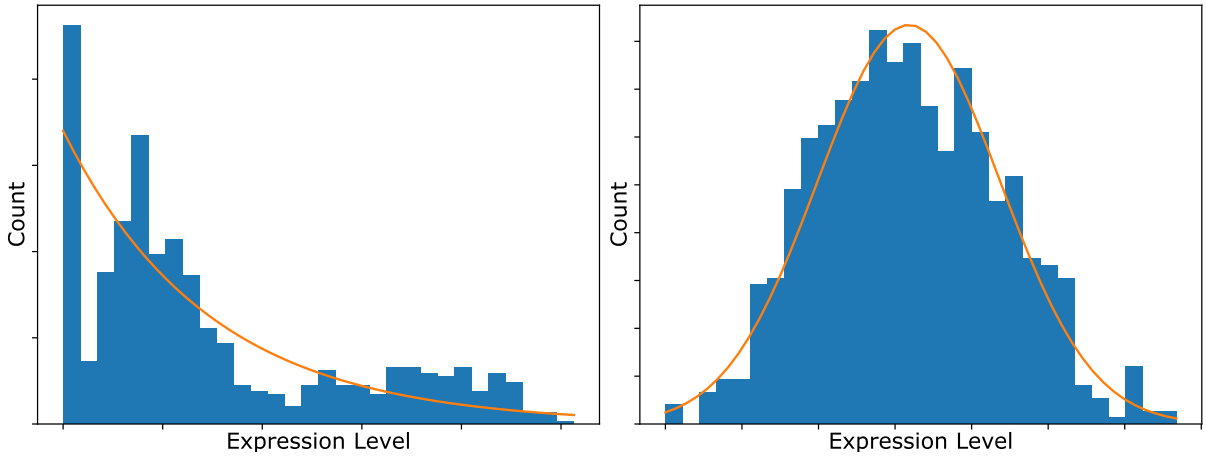
em que $\sup_x$ é o supremo dos conjunto de distâncias.

Uma vez que todos os dados de expressão gênica estão ajustados em uma das duas distribuições, o formato de sino exibido na distribuição normal é um indicativo de que tal gene está diferencialmente expressado. Por outro lado, as distribuições exponenciais mostram que a expressão gênica desse gene está majoritariamente concentrada em valores baixos. As Figuras 39a e 39b mostram exemplos desses dois casos.

Agora, a discretização leva em consideração a distribuição de cada gene. Primeiramente, se o gene foi ajustado na distribuição exponencial, o significado é que existem mais valores próximos de zero do que qualquer outro valor. Por esse motivo, esse gene será discretizado em nível lógico 0 para todos os pontos de *pseudotime*.

Figura 39 – Ajuste de distribuições em dados de expressão gênica.

(a) Distribuição Exponencial em GED. (b) Distribuição Normal em GED.

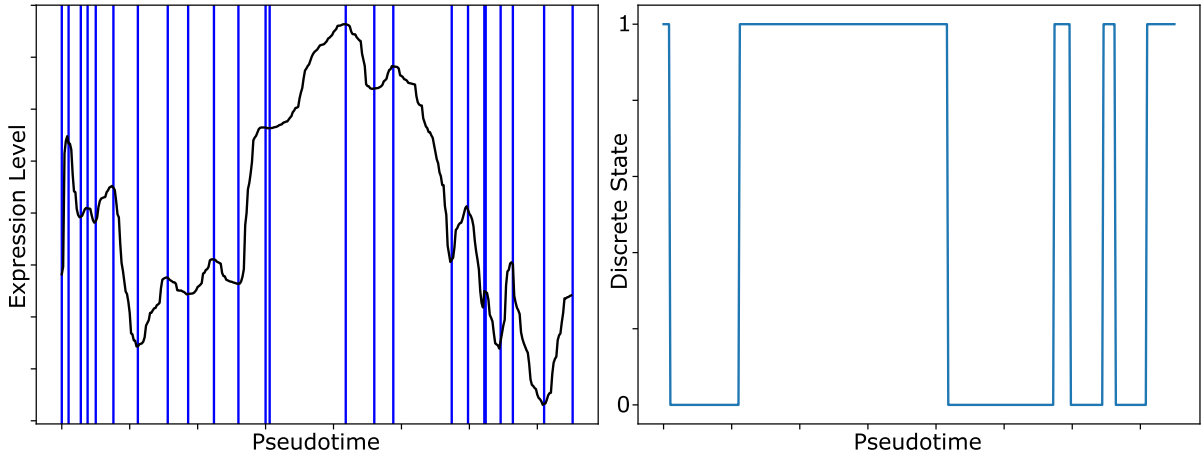


Fonte: SILVA *et al.* (2024).

Se o gene foi ajustado na distribuição normal, é necessário identificar os pontos nos quais os níveis de expressão gênica estão aumentando e aqueles onde os níveis de expressão gênica diminuem. Para isso, o DSSPD divide os dados de expressão gênica em intervalos nos quais o sinal da diferença absoluta entre pontos sucessivos de *spline* muda, conforme exibido na Figura 40a.

Figura 40 – Etapas do DSSPD.

(a) Pontos de corte do DSSPD (linhas azuis verticais) e o *spline* da GED (linha preta). (b) Estados discretos resultantes (0 ou 1).



Fonte: SILVA *et al.* (2024).

Neste caso, o primeiro intervalo determinado pelos pontos de corte do DSSPD terá seu estado discreto atribuído conforme a Equação 3.4. Esta equação é a mesma utilizada pelo método TSD com a diferença de que os dados agora não estão mais normalizados utilizando z-*scores* com distribuição normal. Desta forma, a'_{ij} e $a'_{i(j-1)}$ são pontos de *spline*

sucessivos.

$$a_{ij} = \begin{cases} 1, & a'_{ij} - a'_{i(j-1)} \geq 0, e \\ 0, & a'_{ij} - a'_{i(j-1)} < 0 \end{cases} \tag{3.4}$$

Uma vez que é possível haver mudança de sinal nos dados de *spline* devido a ruídos e fenômenos biológicos tais como a heterogeneidade celular, não é interessante considerar que toda mudança de sinal esteja indicando uma variação da expressão gênica suficientemente considerável a ponto de mudar o estado lógico (de ativação para inibição ou vice-versa). Com isso, os demais intervalos serão discretizados de acordo com sua variância em relação à variância total dos dados de expressão gênica. Dada uma porcentagem total da variância da expressão gênica (μ_{var}), o intervalo atual é discretizado utilizando a Equação 3.4 se $var(GED_{ti:tj}) > \mu_{var} \times var(GED_{t1:tn})$. Caso contrário, o intervalo atual recebe o mesmo estado discreto do intervalo anterior (p_s). Aqui, $GED_{ti:tj}$ é o intervalo atual, iniciando no ponto do tempo *ti* e finalizando no ponto *tj*, $var(GED_{t1:tn})$ é a variância do intervalo atual, e $var(GED_{t1:tn})$ é a variância total dos dados de expressão gênica.

A etapa de separação dos intervalos do DSSPD é apresentado na Figura 40a e os estados discretos resultantes são apresentados na Figura 40b. O Algoritmo 5 resume o procedimento do DSSPD, e um exemplo é apresentado na Figura 41.

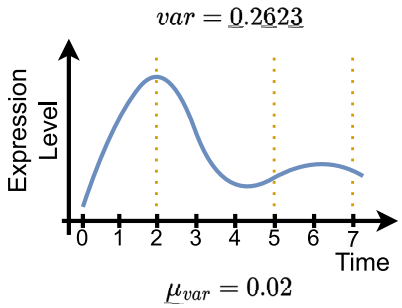
Algoritmo 5: Pseudocódigo do procedimento da DSSPD.

```
Data: GED, distribuição μ_var
Result: estados discretos de um GED dado
begin
    if distribuição = exponencial then
        atribui estado discreto 0 para todos os pontos do GED
    else
        t1 ⟵ primeiro ponto no tempo do GED;
        tn ⟵ último ponto no tempo do GED;
        intervalos ⟵ pontos no tempo com mudança de sinal;
        p_s ⟵ último estado do TSD aplicado no primeiro intervalo;
        for i ← 1 to size(intervalos)-1 do
            ti ⟵ intervalos[i];
            tj ⟵ intervalos[i+1];
            if var(GED[ti:tj] ≤ μ_var × var(GED[t1 : tn]) then
                atribui estado discreto p_s para todos pontos do GED[ti:tj];
            else
                aplica a discretização no intervalo GED[ti:tj];
            end if
        end for
    end if
end
```

Figura 41 – Exemplo simples da aplicação da DSSPD. Primeiramente, DSSPD identifica onde existem aumentos e diminuições dos dados de expressão gênica, conforme apresentado na tabela à esquerda. Então, DSSPD calcula a variação do intervalo no qual o padrão de expressão gênica muda. Se a variância for maior que o produto entre a variância de toda a expressão gênica e o parâmetro μ_{var}, a discretização é aplicada neste intervalo. Caso contrário, o estado discreto anterior é mantido para todo o intervalo atual.

Expression Data

Time Point	Expression Level	Previous	Status
0	0.5	-	
1	1.7	>	
2	2.2	>	
3	1.8	<	Change
4	0.8	<	
5	1.2	>	Change
6	1.4	>	
7	1.3	<	Change



Interval (int)	$var(int)$	$\mu_{var} \bar{\times} var$	$var(int) > \bar{var}$	Result
[3,4]	0.25	0.0052	True	Use TSD
[5,6]	0.01	0.0052	True	Use TSD
[7]	0	0.0052	False	Keep

Discrete States

Time Point	Discrete State
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1

Fonte: SILVA *et al.* (2024).

3.4 INFERÊNCIA DA REDE BOOLEANA VIA PROGRAMAÇÃO GENÉTICA CARTESIANA

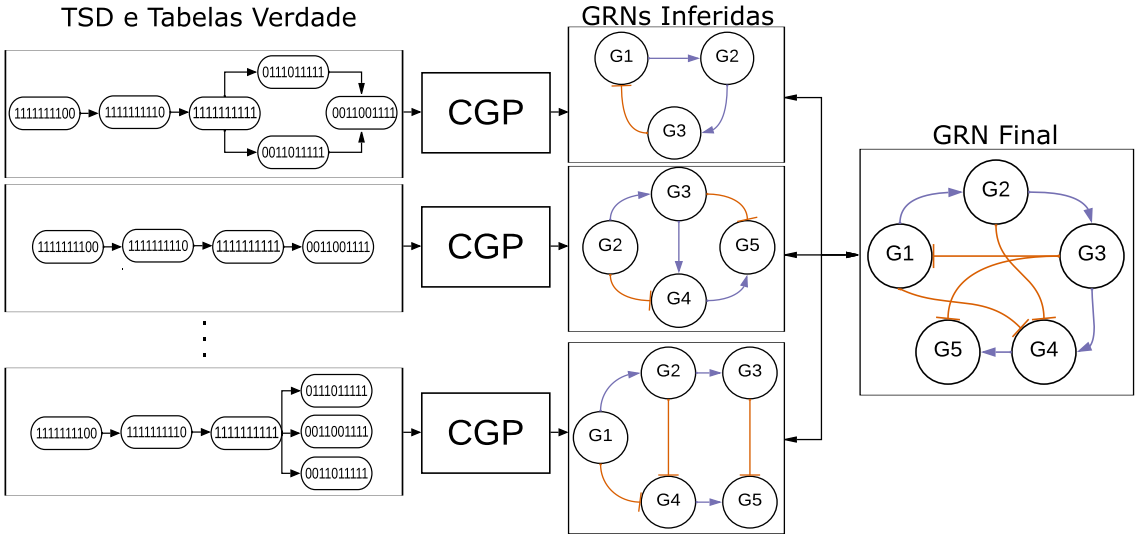
A tabela verdade resultante da etapa anterior (discretização) é utilizada como entrada para a CGP. O objetivo aqui é encontrar um circuito lógico combinacional (CLC - do inglês *combinational logic circuit*), que atenda às transições fornecidas pela tabela verdade. O procedimento geral de inferência da rede de regulação gênica via CGP é apresentado na Figura 42.

O número de combinações de entradas cresce exponencialmente com o número de entradas. Portanto, no pior caso, a avaliação e uma solução candidata da CGP tem uma complexidade computacional de $O(2^{n_i})$, onde n_i é o número de entradas (genes). Contudo, é importante esclarecer que, conforme apresentado na Seção 3.3, as tabelas verdade geralmente não são completas, o que reduz o tempo de processamento. Além disso, conforme apresentado na Seção *2.7.3.2*, esta etapa é paralelizada em GPU e foram realizadas modificações na representação dos indivíduos e operadores de movimento para a evolução de CLCs e a obtenção de GRNs Booleanas na abordagem de SILVA *et al.* (2023).

O circuito então é evoluído usando duas etapas:

1. aumentar o número de acertos em relação à tabela verdade, e
2. reduzir o número de elementos lógicos (portas).

Figura 42 – O modelo Booleano é inferido usando CGP considerando o TSD e a tabela verdade obtida na etapa de discretização. Essa tabela verdade é dada como entrada para CGP que infere, por exemplo, uma GRN para cada gene. Então, considerando todas as GRNs inferidas, a GRN final é aquela obtida mesclando as GRNs parciais.



Fonte: SILVA *et al.* (2023).

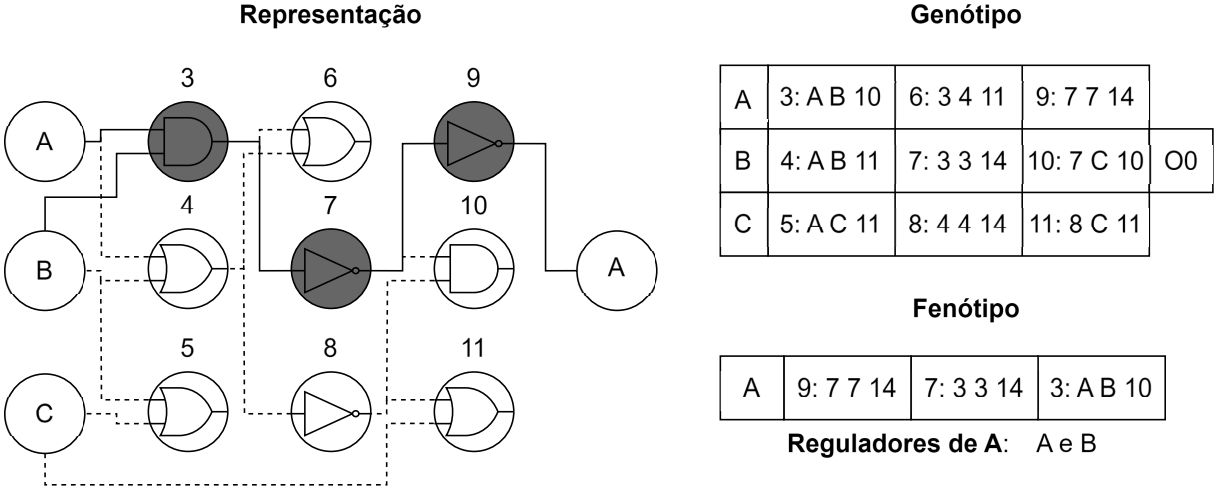
Enquanto o primeiro passo gera uma solução que modela os dados discretizados, a segunda etapa reduz a complexidade do modelo. Reduzir a complexidade do modelo permite a investigação dos principais genes que contribuem para a rede de regulação, e a literatura ressalta que as relações regulatórias de um gene geralmente envolvem uma pequena quantidade de outros genes (CHAN; STUMPF; BABTIE, 2017; WOODHOUSE *et al.*, 2018; LONARDI; SZPANKOWSKI; YANG, 2004). É importante ressaltar que as etapas são realizadas separadamente. Primeiramente objetiva-se encontrar uma solução factível. Quando esta é obtida, o recurso computacional restante é utilizado para minimizar o número de elementos lógicos.

A CGP pode ser utilizada com uma única saída, representando o gene alvo cuja rede de regulação gênica está sendo inferida. Isso é justificado pelo fato de que é mais fácil obter soluções factíveis com saídas únicas devido à simplicidade das tabelas verdade. Em algumas investigações da literatura (SOUZA *et al.*, 2020; SILVA; SOUZA; BERNARDINO, 2019; SILVA; BERNARDINO, 2018), muitas vezes soluções factíveis não são obtidas mas várias saídas já atendem suas respectivas tabelas verdade. Além disso, pode ser de interesse do usuário encontrar possíveis reguladores para um único gene. Contudo, devido ao alto número de situações de irrelevância comumente obtidas no processo de discretização, encontrar uma solução factível pode não ser um problema. Desta forma, é possível utilizar a CGP com múltiplas saídas. Isto consiste em evoluir um único circuito contendo todos os genes como entradas e também como saídas. Neste caso, é possível reaproveitar subcircuitos para mais de um gene. Isso mostrou-se favorável para a obtenção de circuitos menos complexos e mais compactos (SOUZA *et al.*, 2020; SILVA; SOUZA;

BERNARDINO, 2019; SILVA; BERNARDINO, 2018). Essa opção é mais indicada quando deseja-se obter um circuito inteiro e, consequentemente, uma única rede, não havendo a necessidade de unificação das redes parciais. Escolher entre uma única saída ou múltiplas saídas é tarefa do usuário.

No processo de inferência, uma vez obtido o circuito que modele corretamente a tabela verdade fornecida, assume-se como reguladores de um gene (saída), todos aqueles genes da entrada que contribuem para o circuito. Isso é o equivalente a dizer, para a CGP, que os reguladores de uma dada saída são as entradas primárias do fenótipo dessa saída, conforme apresentado na Figura 43.

Figura 43 – Os reguladores de um dado gene, representado na saída do indivíduo da CGP (A), são determinados pelas entradas primárias que compõem o fenótipo (nós ativos em cinza) dessa saída. Desta forma, os reguladores de A são A e B.



Fonte: Elaborado pelo autor (2024).

Independentemente da escolha da forma pela qual as redes serão evoluídas em termo de número de saídas, é importante que sejam realizadas diversas execuções independentes, de cada gene (para saída única) ou de toda a rede (para múltiplas saídas). Isso gera estatísticas sobre as probabilidades de ocorrência das relações regulatórias inferidas. Esse valor de probabilidade é utilizado para ordenar as relações regulatórias, da mais forte para a mais fraca, necessário para a avaliação das redes inferidas tanto no *framework* BEELINE quanto no módulo de avaliação do CGPGRN. Um exemplo disso é apresentado na Tabela 7, considerando 5 execuções independentes para um problema de 5 genes (A,B,C,D e E), cujo gene alvo é A.

Conforme ressaltado anteriormente, no caso de utilizar uma única saída, faz-se necessário montar uma rede completa que represente as relações regulatórias de todos os genes envolvidos inicialmente. O mesmo acontece caso o problema possua mais de um *pseudotime*. Nesse caso, o algoritmo de inferência é executado para cada linhagem

Tabela 7 – Exemplo da geração de probabilidades de relações regulatórias para um gene A. Os genes B e C aparecem em 3/5 execuções e o gene D em 1/5. Isso significa que existe uma probabilidade de 60% que os genes B e C regulem o gene A, enquanto existe uma probabilidade de 20% para regulação de A por D.

Execução	Expressão Lógica	Genes Envolvidos
1	And(B, C)	B,C
2	Or(B, C)	B,C
3	B	B
4	C	C
5	D	D

Fonte: Elaborado pelo autor (2022).

separadamente. Para obter a rede completa, basta unificar as relações obtidas, ordenando-as da mais forte para a mais fraca. Em caso de repetição da relação regulatória, opta-se por aquela mais forte. Já para a representação da rede, basta reproduzir as relações obtidas em cada rede inferida separadamente em uma única rede. A unificação das relações e a rede unificada são apresentadas na Tabela 8 e Figura 44, respectivamente, para uma rede de 5 genes (A, B, C, D e E).

Tabela 8 – Rede final obtida a partir do ranqueamento das relações regulatórias para 5 genes.

Regulador	Regulado	Probabilidade
E	A	1,0
B	C	1,0
D	E	1,0
A	B	0,8
A	D	0,6
C	D	0,4
B	B	0,2

Fonte: Elaborado pelo autor (2022).

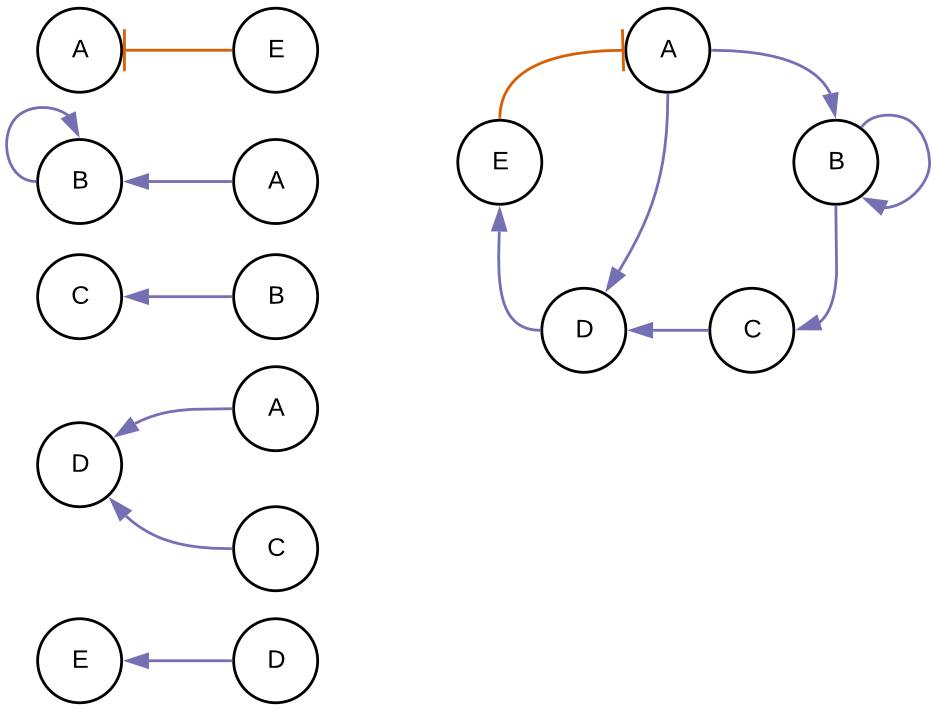
3.5 DETERMINAÇÃO DOS COEFICIENTES NUMÉRICOS DO MODELO CONTÍNUO

A saída da etapa anterior é uma rede Booleana (lógica) que, conforme descrito na Seção 2.11, constituem a forma mais simples de modelagem de GRNs, simplificando a estrutura e a dinâmica de regulação gênica. Esse tipo de modelo provê uma medida qualitativa dos mecanismos regulatórios gênicos. Por outro lado, modelos baseados em EDOs são modelos de GRNs simbólicos acurados, capazes de representar quantitativamente as interações gênicas e possibilitando a predição dos níveis de expressão gênica ao longo

Figura 44 – Obtenção da GRN final. A GRN final é obtida unificando as relações regulatórias obtidas nas GRNs parciais.

(a) Redes regulatórias parciais.

(b) Rede regulatória final.



Fonte: SILVA *et al.* (2020).

do tempo. Desta forma, a última etapa do método proposto consiste em converter um modelo Booleano funcional em um sistema de equações diferenciais ordinárias. Entretanto, os coeficientes numéricos dessas EDOs precisam ser determinados. Isso significa dizer que o conjunto de todas as etapas do método proposto produzem três modelos de GRNs:

1. um modelo booleano qualitativo,
2. um modelo contínuo na forma de um sistema de EDOs com coeficientes numéricos indefinidos, e
3. um modelo final na forma de um sistema de EDOs.

A obtenção do modelo Booleano qualitativo foi descrita nas seções anteriores. Para o modelo contínuo na forma de um sistema de EDOs, a metodologia considerada aqui é baseada na de WITTMANN *et al.* (2009) e apresentada na Seção **2.11.2**.

Consideramos, também, o uso dos HillCubes como função de atualização contínua. Neste caso, cada espécie já no domínio contínuo, é representada por uma função de Hill, conforme apresentado na Equação 2.36. Nesta equação, z determina a inclinação da curva (cooperação de interação) e k é um *threshold*.

O HillCube é a aplicação da função de Hill em cada variável contínua $\overline{x_i}$. Desta forma, a nova função de atualização é dada pela Equação 2.37. Por fim, o comportamento temporal é dado pela Equação 2.40 onde $\overline{B}$ é a função de atualização contínua (HillCube), que descreve a produção das espécies X_i e um termo de decaimento de primeira ordem e τ_i é o tempo de vida das espécies X_i.

Desta forma, o modelo contínuo obtido contém coeficientes numéricos (z, k e τ) que precisam ser determinados. Para esta tarefa, utiliza-se aqui as Estratégias Evolutivas, descritas na Seção **2.7.2**.

Sendo assim, cada solução candidata na ES contém os parâmetros z, k e τ de todos os HillCubes presentes no modelo contínuo. Outro fato importante é que a complexidade do problema de otimização é diretamente proporcional ao tamanho do sistema de EDOs. O problema de otimização consiste em determinar $\mathbf{g}(\mathbf{x}, t)$ tal que $\mathbf{x}'(t) = \mathbf{g}(\mathbf{x}, t)$ se adeque aos dados observados $(\mathbf{x}_i, t_i)$, com $i = 1, \ldots, m$. Esse problema foi discutido na Seção 2.5, onde um modelo dinâmico pode ser avaliado através da integração numérica do sistema de EDOs $\mathbf{x}'(t) = \mathbf{g}(\mathbf{x}, t)$, correspondendo ao candidato $\mathbf{g}(\mathbf{x}, t)$. Neste trabalho opta-se pelo uso da integração numérica devido ao fato desta abordagem tender a gerar modelos mais acurados (BERNARDINO; BARBOSA, 2011, 2010). Além disso, o modelo candidato é integrado como um todo e depois o resultado entre o modelo e o dado real é comparado. Já a função objetivo é a soma dos módulos das diferenças para cada ponto no tempo.

Conforme ressaltado na Seção **2.11.2**, os dados originais são normalizados no intervalo [0, 1]. Além disso, o modelo candidato é integrado a partir de um valor inicial conhecido. Isso gera um conjunto de valores aproximados. Por fim, os valores aproximados são comparados com esses valores esperados normalizados. O resultado em termos de erro do modelo é, portanto, a soma em módulo das diferenças entre os valores esperados e previstos.

Outro fator importante a ser destacado sobre essa etapa é que ela é um mecanismo adicional que enriquece a proposta, como apresentado na Figura 27. Se for da vontade do experimentalista e/ou do usuário obter somente uma rede Booleana, para obter informações qualitativas de um sistema biológico, como por exemplo grupos de genes que estão relacionados em alguma atividade regulatória ou possíveis ativadores e inibidores de um gene alvo específico, esta etapa não precisa ser utilizada.

4 EXPERIMENTOS COMPUTACIONAIS

Experimentos computacionais foram realizados para avaliar a qualidade das redes inferidas pelo método proposto e a reprodutibilidade da dinâmica dos produtos gênicos obtida através do modelo contínuo. Neste capítulo são discutidos os problemas investigados tanto nos experimentos computacionais do método proposto quanto das investigações e avaliações das etapas do método proposto, configurações dos algoritmos e apresentação e discussão dos resultados obtidos. Os experimentos são divididos em 3 grupos: (i) avaliação do método proposto em problemas *benchmark*, contendo problemas sintéticos, acurados e dados experimentais, (ii) avaliação do método proposto para modelos contínuos, onde consideram-se dados de *Drosophila*, e (iii) avaliação das etapas do método proposto, onde discutem-se estudos realizados sobre o impacto do pré-processamento, da discretização, do agrupamento e da inferência do modelo Booleano.

O código fonte do *framework* CGPGRN, manual de uso e exemplos estão disponíveis publicamente[1].

O restante deste capítulo é dividido como segue. Na Seção 4.1 são apresentados todos os problemas utilizados para a avaliação da proposta. Na sequência, os resultados obtidos pela proposta em problemas *benchmark* são apresentados na Seção 4.2. As comparações entre os resultados obtidos pela proposta em organismos amplamente estudados na literatura são apresentadas na Seção 4.3. Na Seção 4.4, compara-se a proposta com outros métodos Booleanos e baseado em metaheurísticas para a inferência de GRNs. A avaliação do método proposto para modelos contínuos é apresentada na Seção 4.5 e, por fim, a Seção 4.6 discute e explora as principais etapas do método proposto. Além disso, conclusões parciais sobre os experimentos realizados são apresentadas na Seção 4.8.

4.1 DESCRIÇÃO DOS PROBLEMAS ABORDADOS

Neste trabalho, utiliza-se quatro grupos de dados, abrangendo problemas *benchmark* comumente utilizados na literatura, dados simulados de modelos de ritmos circadianos com 5 e 10 espécies, corroborados e amplamente difundidos, dados de expressão gênica de organismos amplamente estudados, tais como a *E. coli* e a *S. cerevisiae*, a fim de demonstrar a possibilidade de exploração dos fenômenos biológicos envolvidos e subredes de *E. coli* utilizadas na comparação com métodos de inferência Booleanos e baseados em metaheurística.

Para o primeiro grupo são considerados os problemas *benchmark* propostos por PRATAPA *et al.* (2020), que podem ser divididos em três grandes classes: sintéticos, acurados e experimentais.

[1] https://github.com/jeduaardo/cgpgrn

Os problemas sintéticos foram gerados com dois propósitos: (i) utilizar uma GRN que pode servir como *ground-truth*, e (ii) ter conjuntos de dados de expressão gênica baseados em *single-cell* que estão isolados de qualquer limitação introduzida pelos algoritmos de inferência de *pseudotime*. Desta forma, seis redes sintéticas foram geradas. Os problemas, quantidade de genes e quantidade de *pseudotimes* são apresentados na Tabela 9.

Tabela 9 – Problemas sintéticos com seus respectivos número de genes e *pseudotimes*.

Problema	#Genes	#*Pseudotimes*
Linear	7	1
Cycle	6	1
Linear Long	18	1
Bifurcating	7	2
Bifurcating Converging	10	2
Trifurcating	8	3

Fonte: Elaborado pelo autor (2022).

Segundo PRATAPA *et al.* (2020), os problemas sintéticos podem ser definidos como:

1. *Linear*: uma ativação gênica em cascata que resulta em uma única trajetória temporal com estados inicial e final distintos,
2. *Linear Long*: similar à linear mas com um número maior de genes intermediários,
3. *Cycle*: um circuito de oscilação que produz uma trajetória linear onde o estado final sobrepõe-se ao estado inicial,
4. *Bifurcating*: uma rede que contém um motivo de inibição mútua entre dois genes resultando em dois ramos distintos a partir de uma trajetória comum,
5. *Trifurcating*: motivos de inibição mútua envolvendo três genes nesta rede resultar em três estados estacionários distintos, e
6. *Bifurcating Converging*: uma bifurcação inicial cria dois ramos, que finalmente convergem para um único estado estacionário.

Diferentemente da maioria dos estudos recentes em inferência de GRNs (GAO *et al.*, 2018; SANCHEZ-CASTILLO *et al.*, 2018; CHAN; STUMPF; BABTIE, 2017; LIM *et al.*, 2016; CHEN; MAR, 2018), em PRATAPA *et al.* (2020) optou-se por não utilizar o *GeneNetWeaver* (SCHAFFTER; MARBACH; FLOREANO, 2011) para criar os dados sintéticos de *single-cell* pois não haviam trajetórias discerníveis nas projeções 2D desses dados. Desta forma, utilizou-se o BoolODE (PRATAPA *et al.*, 2020). Para cada gene em

uma GRN, o BoolODE requer uma função Booleana que especifica como os reguladores daquele gene controlam seu estado que servem como base para a geração de uma equação diferencial ordinária não linear. Além disso, adicionou-se termos de ruído para tornar a equação estocástica (SAELENS *et al.*, 2019; OCONE *et al.*, 2015). Para cada problema, cinco grandes conjuntos são criados, contendo 100, 200, 500, 2.000 e 5.000 células. Além disso, para cada um desses conjuntos, 10 diferentes conjuntos de dados são disponibilizados. Isso resulta em 50 diferentes conjuntos de dados de expressão gênica por problema.

Os problemas acurados foram gerados pois sub-redes densas de GRNs de grande escala que foram utilizadas para gerar o conjunto de dados sintéticos podem não capturar a regulação complexa em nenhum processo de desenvolvimento específico. Por isso, quatro modelos Booleanos publicados: desenvolvimento da área cortical de mamíferos (mCAD - *mammalian cortical area development*) (GIACOMANTONIO; GOODHILL, 2010), desenvolvimento da medula espinhal ventral (VSC - *ventral spinal cord development*) (LOVRICS *et al.*, 2014), diferenciação de células-tronco hematopoiéticas (HSC - *hematopoietic stem cell*) (KRUMSIEK *et al.*, 2011) e determinação de sexo gonadal (GSD - *gonadal sex determination*) (RÍOS *et al.*, 2015) foram utilizados juntamente com o BoolODE para criar dez conjuntos de dados diferentes com 2.000 células para cada modelo. Além disso, para cada conjunto de dados, três configurações de *dropouts* (0%, 50% e 70%) estão disponíveis, cada um com 10 variações. Isso resulta em 30 conjuntos de dados de expressão gênica para cada problema acurado. Os *pseudotimes* foram inferidos usando *Slingshot* (STREET *et al.*, 2018). Os problemas, quantidade de genes e quantidade de *pseudotimes* são apresentados na Tabela 10.

Tabela 10 – Problemas acurados com seus respectivos número de genes e *pseudotimes*.

Problema	#Genes	#*Pseudotimes*
mCAD	5	2
VSC	8	5
HSC	11	4
GSD	19	2

Fonte: Elaborado pelo autor (2022).

Por fim, cinco conjuntos de dados scRNA-Seq experimentais foram obtidos da literatura, dois em células humanas e três em células de camundongos, composto por 7 tipos celulares. Os dados foram pré-processados de acordo com o especificado em suas respectivas publicações. Maiores informações são encontradas em (PRATAPA *et al.*, 2020). Os problemas experimentais são denominados hESC, hHep, mDC, mESC, mHSC-E, mHSC-GM e mHSC-L.

O hESC (CHU *et al.*, 2016) é um conjunto de dados a partir de um experimento de scRNA-Seq de curso no tempo derivado de 758 células ao longo do protocolo de

diferenciação para produzir células endodérmicas definitivas a partir de células tronco embrionárias humanas, medidas em 0, 12, 24, 36, 72 e 96h.

O hHep (CAMP *et al.*, 2017) é derivado de um experimento scRNA-Seq em células tronco pluripotentes induzidas (iPSCs - do inglês *induced pluripotent stem cells* em uma cultura bidimensional diferenciando para células semelhantes a hepatócitos. Esse conjunto de dados contém 425 medidas de múltiplos pontos no tempo: dias 0 (iPSCs), 6, 8, 14, e 21 (células semelhantes a hepatócitos maduras).

Já o mDC (SHALEK *et al.*, 2014) utiliza células dendríticas de camundongo, com mais de 1.700 células dendríticas derivadas de medula óssea sob diversas condições.

O conjunto de dados mESC (HAYASHI *et al.*, 2018) (células tronco embrionárias - do inglês *embryonic stem cell*) contém medidas de expressão de 421 células endodérmicas primitivas (PrE - do inglês *primitive endoderm*) diferenciadas a partir de mESCs, coletadas em cinco pontos do tempo diferentes: 0, 12, 24, 48 até 72h.

Por fim, os mHSCs (NESTOROWA *et al.*, 2016) são derivados de dados de expressão gênica normalizadas de 1.656 células-tronco hematopoéticas e células progenitoras (HSPCs - do inglês *hematopoietic stem and progenitor cells*). Os diferentes sufixos representam as diferentes linhagens obtidas no *pseudotime*, a saber eritróide (E), granulócito-monócito (GM) e linfóide (L).

Cada um dos problemas experimentais é disponibilizado em 4 configurações diferentes com variadas quantidades de espécies envolvidas. Consideram-se para tal, um número base de genes (500 ou 1000) e a inclusão, ou não, de todos os fatores de transcrição significativamente variantes para esses genes. Desta forma, as quatro configurações resultantes são 500nTF, 500TF, 1000nTF e 1000TF, onde os sufixos nTF e TF indicam não considerar ou considerar os fatores de transcrição, respectivamente. Com isso, os conjuntos de dados na configuração nTF possuirão sempre 500 ou 1000 espécies enquanto as configurações TF possuirão uma quantidade maior de espécies[2]. Além disso, existe somente um *pseudotime* para os problemas experimentais. Os problemas experimentais e o número máximo de genes disponíveis em cada um dos conjuntos de dados são sumarizados na Tabela 11.

Como não existem redes *ground-truth* para os problemas experimentais, três diferentes redes de referência são consideradas: *cell-type-specific ChIP-Seq*, *NonSpecific ChIP-Seq* e redes de interações funcionais (STRING), conforme discutido na Seção 2.12.

Já para o segundo grupo, dois conjuntos de dados foram gerados a partir de duas redes de regulação gênicas cuja dinâmica é descrita por sistemas de equações diferenciais. Ambas as EDOs modelam a rede do ritmo circadiano da *Drosophila*. O ritmo circadiano designa o período no qual o ciclo biológico de praticamente todos os seres vivos é baseado, influenciado principalmente pela variação da luz, temperatura, marés e ventos. Assim,

[2] Esse fato não é observado para o problema mHSC-L e é discutido na Seção 4.7.

Tabela 11 – Problemas experimentais com seus respectivos número de genes. O prefixo *h* indicam células humanas enquanto o *m*, camundongos.

Problema	#Genes
hESC	17.735
hHep	11.515
mDC	7.371
mESC	18.385
mHSC-E	4.762
mHSC-GM	4.762
mHSC-L	4.762

Fonte: Elaborado pelo autor (2022).

cada conjunto de dados corresponde à concentração de espécies da rede que é repetida em um ciclo de 24 horas.

Um dos sistemas de EDOs possui cinco variáveis de estado (portanto, cinco EDOs) e o outro possui dez variáveis de estado. O modelo de cinco variáveis é uma GRN que foi proposta para oscilações circadianas na proteína PER de *Drosophila* e seu respectivo gene e mRNA (GOLDBETER, 1995). A representação e explicação do modelo são apresentadas na Figura 45.

O modelo de dez variáveis leva em consideração as oscilações circadianas na *Drosophila* envolvendo a regulação negativa da expressão gênica pela PER e TIM (LELOUP; GOLDBETER, 1998). O modelo esquemático e sua descrição são apresentados na Figura 46.

Mais detalhes sobre esses conjuntos de dados podem ser obtidos em (GOLDBETER, 1995) e LELOUP; GOLDBETER (1998).

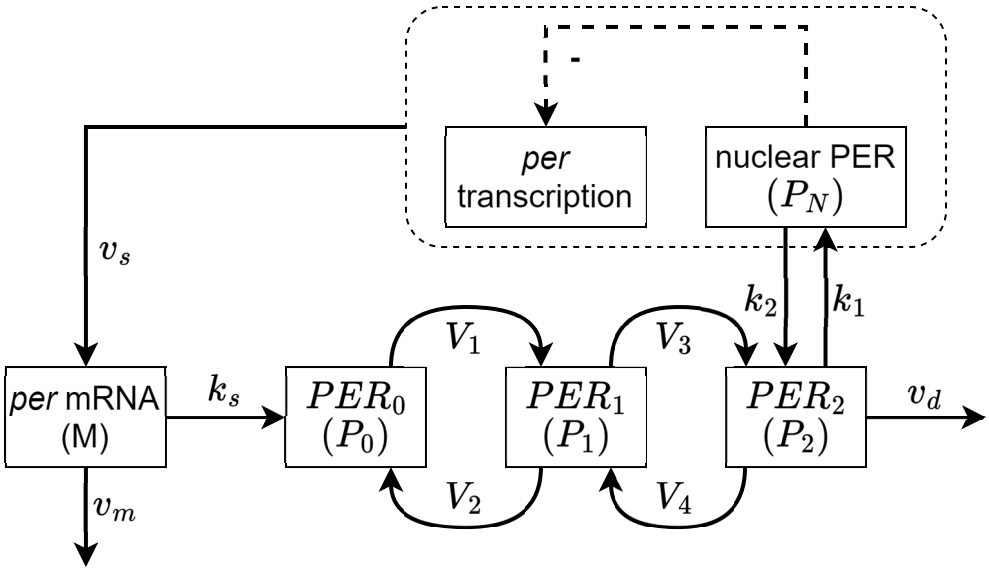
Os sistemas de EDOs de ambos os modelos, disponíveis nos Anexos A e B, para os modelos de 5 e 10 espécies, respectivamente, foram resolvidos através do solucionador MATLAB® ODE, resultando em uma série temporal contento 50 pontos uniformemente espaçados em um intervalo de tempo de simulação de 0 a 72 horas.

O terceiro conjunto de dados abrange organismos amplamente estudados na literatura, tais como a *E. coli* e a *S. cerevisiae*.

O primeiro *dataset* representa o processo de reparo do DNA da *E. coli*, conhecido como *E. coli SOS repair*. Este *dataset* é composto por 8 genes (*uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, e polB*, conforme apresentado na Figura 47.

Este *dataset* é composto por dados reais de expressão gênica de *E. coli* e a estrutura da sua rede tem sido verificada por experimentos reais (SHEN-ORR *et al.*, 2002; RONEN *et al.*, 2002). Para tal, 4 experimentos foram realizados, mensurando os 8 genes em um total de 5 amostras. Com isso, quatro *datasets* são gerados.

Figura 45 – Modelo do ritmo circadiano de 5 espécies para oscilações na PER e *per* mRNA. O *per* mRNA (M) é sintetizado no núcleo e transferido para o citosol, onde acumula-se em uma taxa máxima v_s; lá, é degradada por uma enzima de taxa máxima v_m e uma constante de Michaelis K_m. A taxa de síntese da proteína PER, proporcional a M, é caracterizada por uma taxa aparente constante de primeira ordem k_s. Os parâmetros V_i e K_i (i = 1, $\cdots$ 4) denotam a taxa máxima da constante de Michaelis para a(s) quinase(s) e fosfatase(s) envolvidas na fosforilação reversível de P_0 em P_1 e P_1 em P_2, respectivamente. A forma totalmente fosforilada (P_2) é degradada por uma enzima de taxa máxima v_d e constante de Michaelis K_d, e transportada para o núcleo a uma taxa caracterizada pela constante de velocidade aparente de primeira ordem k_1. O transporte da forma nuclear bifosforilada de PER (P_N) para o citosol é caracterizado pela constante de velocidade aparente de primeira ordem k_2. O *feedback* negativo exercido pelo PER nuclear por transcrição é descrito por uma equação do tipo Hill.



Fonte: Adaptado de GOLDBETER (1995).

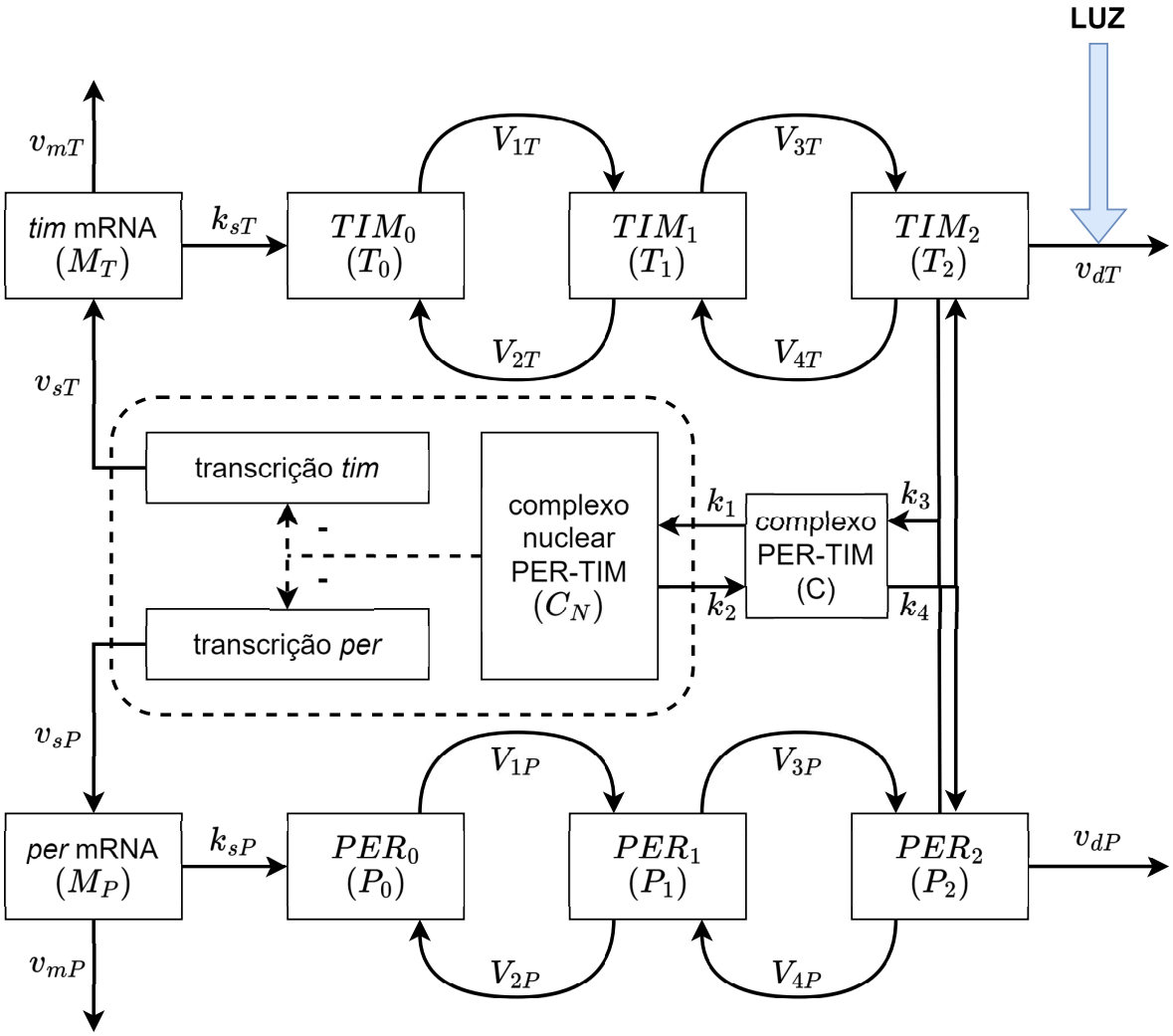
O segundo *dataset* considerado aqui é o IRMA (*In vivo Reverse-engineering and Modeling Assessment*), que é uma rede sintética *in vivo* da vida real construída dentro da levedura *Saccharomyces cerevisiae* (CANTONE *et al.*, 2009). Este *dataset* é amplamente utilizado na literatura para a avaliação do desempenho de GRNs (PIRGAZI; KHANTEY-MOORI, 2018; PENFOLD; WILD, 2011; YANG *et al.*, 2018). Essa rede de pequena escala é composta de cinco genes (CBF1, GAL4, SWI5, GAL80, ASH1) com um total de 8 relações regulatórias, apresentada na Figura 48.

Os últimos *datasets* para este grupo são aqueles da competição DREAM4 (*DREAM 4 - In Silico Network Challenge*[3]).

As topologias das redes foram obtidas a partir da extração de sub redes de redes de regulação transcricional de *E. coli* e de *S. cerevisiae*. A extração é adaptada para incluir preferencialmente partes da rede com ciclos. Relações de autorregulação são removidas.

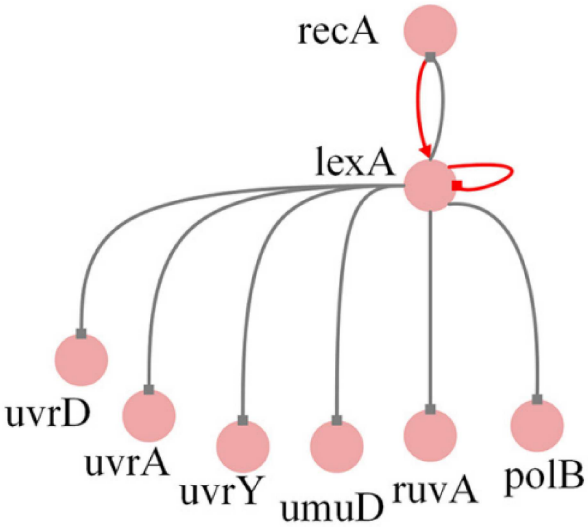
[3] `https://www.synapse.org/#!Synapse:syn3049712/wiki/74630`

Figura 46 – Modelo esquemático das oscilações circadianas na *Drosophila* com 10 espécies envolvendo a regulação negativa da expressão gênica pela PER e TIM. Os mRNAs *per* (M_P) e *tim* (M_T) são sintetizados no núcleo e transferidos para o citosol, onde acumulam-se em taxas máximas v_{sP} e v_{sT}, respectivamente. Lá são degradas enzimaticamente em taxas máximas, v_{mP} e v_{mT}, com constantes Michaelis K_{mP} e K_{mT}. As taxas de síntese das proteínas PER e TIM, proporcionais respectivamente a M_P e M_T são caracterizadas pela taxa constante aparente de primeira ordem k_{sP} e k_{sT}. Os parâmetros V_{iP} (V_{iT}) e K_{iP} (K_{iT}) (i = 1, $\cdots$ 4) denotam a taxa máxima e a constante Michaelis das quinase(s) e fosfatase(s) envolvidas na fosforilação reversível de P_0 (T_0) em P_1 (T_1) e P_1 (T_1) em P_2 (T_2), respectivamente. As formas totalmente fosforiladas (P_2 e T_2) são degradadas por enzimas de taxa máxima v_{dP} e v_{dT} e constantes Michaelis K_{dP} e K_{dT} e formam reversivelmente um complexo C (associação e dissociação são caracterizadas pelas constantes de taxa k_3 e k_4), que são transportadas para dentro do núcleo numa taxa caracterizada pelo taxa constante aparente de primeira ordem k_1. O transporte da forma nuclear do complexo PER-TIM (C_N) para o citosol é caracterizado pela taxa aparente constante de primeira ordem k_2. O *feedback* negativo exercido pelo complexo nuclear PER-TIM nas transcrições da *per* e da *tim* são descritas por uma equação do tipo Hill.



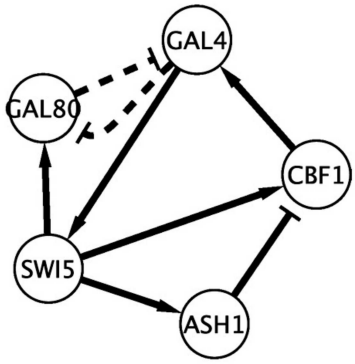
Fonte: Adaptado de LELOUP; GOLDBETER (1998).

Figura 47 – Rede biológica do processo de reparo do DNA da *E. coli*.



Fonte: LEI *et al.* (2023).

Figura 48 – Rede biológica IRMA.



Fonte: CANTONE *et al.* (2009).

Os modelos dinâmicos das redes foram simulados usando um modelo cinético detalhado de regulação gênica. Tanto a transcrição quanto a regulação são modeladas. Contudo, a concentração das proteínas não estão incluídas nos *datasets*. Os *datasets* correspondem aos níveis de concentração de mRNA. Além disso, as simulações são baseadas em equações diferenciais estocásticas (Equações Langevin) para modelar ruídos internos na dinâmica das redes. Todas as redes e dados foram geradas com a versão 2.0 do GeneNetWeaver. Para os experimentos realizados neste trabalho foram considerados os conjuntos de dados do desafio 10 e 100. Para cada um dos desafios, 5 conjuntos de dados diferentes são disponibilizados.

Por fim, o quarto conjunto de dados contempla redes sintéticas geradas a partir de redes de referência de *E. coli*, apresentados em (PUŠNIK *et al.*, 2022), para a comparação com o método ATEN. A geração de dados sintéticos é composta de múltiplas etapas. Primeiramente, conjuntos de GRNs *ground-truth* são gerados a partir de uma rede de

referência de *E. coli* (SCHAFFTER; MARBACH; FLOREANO, 2011; GAMA-CASTRO *et al.*, 2010). As redes geradas servem como base para a geração de modelos cinéticos dinâmicos, que são então simulados para produzir dados de expressão gênica na forma de séries temporais. Além disso, utiliza-se o GeneNetWeaver (SCHAFFTER; MARBACH; FLOREANO, 2011) para importar ou extrair GRNs maiores. Em específico, para os dados de *E. coli*, 10 diferentes conjuntos de dados foram gerados para cada quantidade de genes (16, 32 e 64), resultando em 30 *datasets*.

Todos os problemas utilizados neste trabalho estão publicamente disponíveis no repositório do GitHub[4].

4.2 AVALIAÇÃO DO MÉTODO PROPOSTO EM PROBLEMAS BENCHMARK

Nesta seção são apresentados os resultados referentes à aplicação do CGPGRN nos problemas *benchmark* sintéticos e acurados e nos dados experimentais e suas comparações com os resultados obtidos pelos algoritmos estado da arte apresentados em (PRATAPA *et al.*, 2020) e discutidos na Seção 2.11.

4.2.1 Problemas Sintéticos e Acurados

Para os problemas sintéticos e acurados o pré-processamento é aplicado com valor de suavização determinado através de *shuffle split*, que consiste em um validador cruzado de permutação aleatória, com 10 *splits* e valores de suavização $p \in \{0, 0.05, 0.1, \ldots, 0.95, 0.99\}$. Desta forma, o melhor valor de suavização é aquele que reduz a soma do erro quadrático entre os dados reais e o valor do *spline* para cada ponto. Além disso, adota-se 70% para treinamento e 30% para teste com 10 execuções independentes. Não foi utilizada a etapa de agrupamento, uma vez que tanto para os problemas sintéticos quanto para os acurados as espécies apresentadas são exatamente aquelas disponíveis nas redes *ground-truth* ou já são as espécies necessárias para a modelagem do fenômeno biológico em questão. A discretização foi feita através da ferramenta Gene Expression Data Pre-Processing Tool (GEDPROTOOLS) (GALLO *et al.*, 2015)[5] com o método Bikmeans para todos os problemas exceto o *Cycle*, no qual o método TSD foi usado pois obteve melhores resultados para dados cíclicos em estudos anteriores (SILVA *et al.*, 2020).

Na CGP, foram realizadas 5 execuções independentes para cada gene com um máximo de 20.000 avaliações da função objetivo, uma vez que para esse número de avaliações o processo de busca já não apresentava melhorias, conforme experimentos preliminares. Desta forma, a CGP é executada 5 × N, onde N é o número de genes do problema. Os demais parâmetros da CGP são $n_r = 1$, $n_c = 100$, lb $= n_c$, o conjunto de funções Γ = {AND, OR, NOT, XOR} e o operador de mutação SAM. Os parâmetros

[4] `https://github.com/jeduaardo/tese`
[5] `http://lidecc.cs.uns.edu.ar/files/gedprotools.zip`

da CGP foram determinados de forma empírica, através de estudos preliminares. O conjunto de funções também é resultado de estudos preliminares e contém as portas lógicas básicas, com as quais qualquer CLC pode ser construído (AND, OR e NOT) e a XOR foi introduzida a fim de facilitar a obtenção de expressões do tipo $A\overline{B} + \overline{A}B$ que necessitariam de 5 portas lógicas originalmente.

Para comparação, foram utilizados os algoritmos GENIE3, GRISLI, GRNBOOST2, GRNVBEM, LEAP, PIDC, PPCOR, SCINGE, SCNS, SCODE e SINCERITIES, apresentados em (PRATAPA *et al.*, 2020).

Nesta versão do *framework* a avaliação dos indivíduos da CGP ainda não era realizada em paralelo na GPU. O ambiente computacional utilizado consiste de um processador Intel® i7-4790K, 16 GB DDR3 e GPU GTX760Ti com 2GB. Como discutido na Seção 3.4, para problemas com mais de um *pseudotime*, a CGP é executada independentemente para cada *pseudotime* e a GRN final é a união das GRNs parciais, ordenada pela força das relações regulatórias obtidas.

A Figura 49 apresenta os PPs considerando os melhores valores de AUPRC e AUROC obtidos pelo *framework* BEELINE, e a Figura 50 mostra os PPs considerando os valores de mediana de AUPRC e AUROC. Para facilitar a legibilidade, BEELINE AUPRC e BEELINE AUROC se referem aos valores obtidos pela avaliação das redes inferidas pelo *framework* BEELINE. AUPRC e AUROC dizem respeito à área sob a curva dos PPs, respectivamente, considerando os valores de BEELINE AUPRC e BEELINE AUROC.

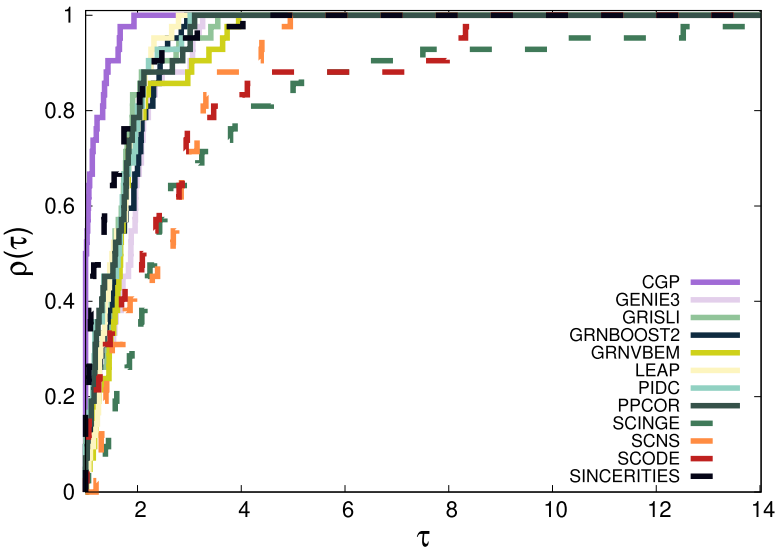
Considerando ambos os cenários, para AUPRC e AUROC, CGP obteve os melhores resultados para a maioria dos problemas (maior $\rho(1) = 0.47$) e o melhor desempenho geral (maior área sob as curvas do PP). Além disso, CGP é a abordagem mais confiável, pois obteve o menor τ tal que $\rho(\tau) = 1$.

Resultados tabulares adicionais para todos os problemas, considerando o melhor valor, primeiro e terceiro quartis, mediana, média, desvio padrão, *boxplots* e as melhores redes reconstruídas são apresentados no material suplementar disponível no repositório. Investigou-se também a significância estatística das diferenças entre os resultados obtidos. Os testes de Kruskal-Wallis (para mediana) e Dunn (para testes *post-hoc*) são os testes estatísticos não paramétricos utilizados aqui. Quando o p-valor ≤ 0.05, conclui-se que a hipótese nula é rejeitada (os resultados são estatisticamente diferentes). Esses resultados também estão disponíveis no repositório. Adicionalmente, o número de problemas no qual um método obteve os melhores resultados em relação à mediana (ou seus resultados são estatisticamente similares aos melhores) é mais uma forma de indicar qual método é o melhor, conforme apresentado na Tabela 12 e discutido mais adiante.

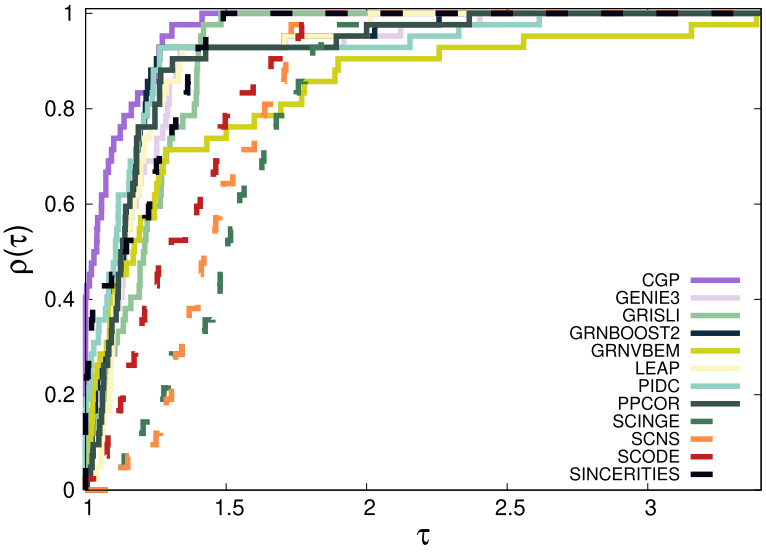
A estratégia proposta que usa a CGP é a melhor ou obteve resultados competitivos quando comparada aos métodos estado da arte, considerando os problemas sintéticos. Em relação aos problemas acurados, como relatado na literatura, quando a taxa de *dropout*

Figura 49 – *Performance Profiles* do melhor para os 12 algoritmos.

(a) AUPRC do melhor.



(b) AUROC do melhor.

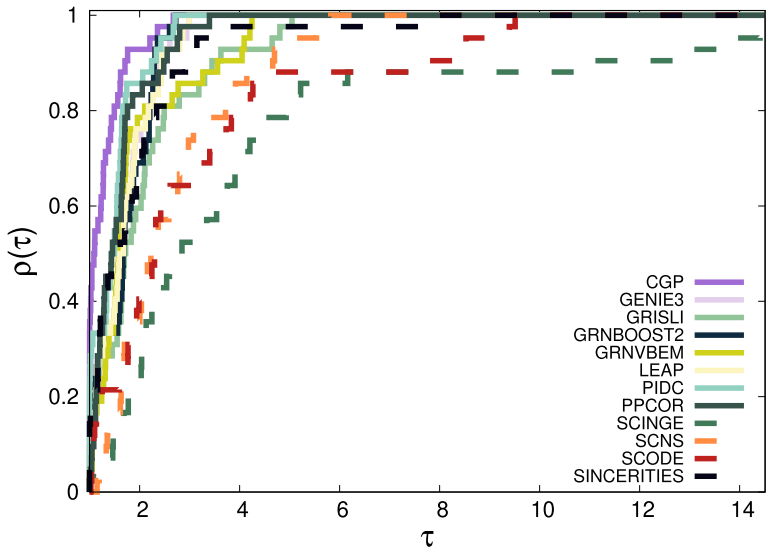


Fonte: da Silva et al. SILVA *et al.* (2023).

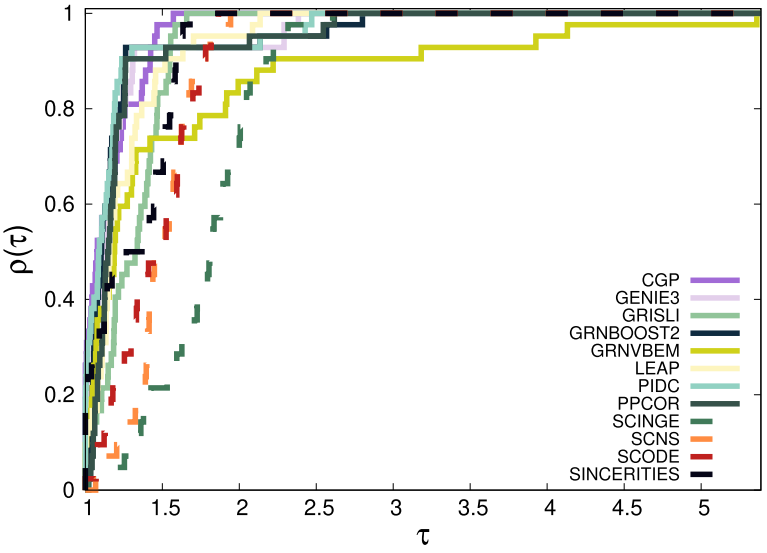
aumenta, o desempenho dos algoritmos diminui. Isso é justificado pelo fato de que a grande presença de valores zero nos dados de expressão gênica dificultam a obtenção de relações regulatórias corretas. Além disso, no caso específico da CGP, a etapa de pré-processamento com o *spline* tem menos pontos para se ajustar. Ainda, a CGP apresenta diferença estatística em 8 dos 11 algoritmos de inferência. Em geral, essa diferença estatística não é observada quando os resultados obtidos pela proposta são comparados com aqueles alcançados com GRISLI, SINCERITIES e SCNS. É possível que a diferença estatística não seja observada quando comparados a esses três algoritmos pois, assim como a CGP, todos utilizam informação pseudotemporal para realizar suas predições.

Figura 50 – *Performance Profiles* da mediana para os 12 algoritmos.

(a) AUPRC da mediana.



(b) AUROC da mediana.



Fonte: SILVA *et al.* (2023).

A Tabela 12 apresenta o número de problemas em que as técnicas obtiveram os melhores resultados para AUPRC e AUROC (os melhores resultados estão em negrito). O problema é contado para cada método no qual não existe diferença estatística observada (p-valor > 0.05) em relação ao melhor.

De acordo com a Tabela 12 é possível perceber que a estratégia proposta obteve os melhores resultados na maioria dos problemas (33 problemas) para BEELINE AUPRC. Quando considerando BEELINE AUROC, a estratégia proposta é o segundo melhor método com maior quantidade de melhores resultados (29 problemas). PIDC obteve o maior número de melhores resultados com significância estatística considerando BEELINE

Tabela 12 – Número de problemas nos quais cada técnica obteve os melhores resultados.

Método	*CGP*	*SCNS*	*PIDC*	*GRNVBEM*	*GENIE3*	*GRNBOOST2*	*PPCOR*	*SCODE*	*SINCERITIES*	*LEAP*	*GRISLI*	*SCINGE*
AUPRC	**33**	2	26	16	9	11	21	5	15	15	11	3
AUROC	29	0	**31**	10	19	23	20	1	13	20	15	0

Fonte: SILVA *et al.* (2023).

AUROC (31 problemas). Entretanto, a saída do PIDC é somente uma lista ordenada das possíveis relações gênicas. As redes na forma de grafos são geradas a partir de *software* de terceiros (GeneNetWeaver). Não existe nenhuma expressão matemática, regra lógica ou qualquer modelo que represente a rede inferida. Além disso, o PIDC não fornece o sinal da regulação (ativação ou inibição). Desta forma, as redes inferidas são sempre apresentadas na forma de redes de coexpressão. A estratégia proposta, além de gerar regras lógicas e ser possível obter um sistema de EDOs, fornece o sinal da regulação. Além disso, é possível reconstruir diretamente as GRNs na forma de grafos.

A Figura 51 apresenta a rede *ground-truth* e a rede reconstruída para o problema mCAD, considerando 0% de *dropout*. A rede original do mCAD tem 5 espécies (Pax6, Fgf8, Emx2, Coup e Sp8) e 14 interações. A CGP foi capaz de obter corretamente a maior parte das relações regulatórias, como apresentado em (GIACOMANTONIO; GOODHILL, 2010). Entretanto, *Sp8* sempre ativa *Fgf8* e essa relação regulatória não foi encontrada pelo nosso método nesse momento. Ao analisar as tabelas verdade geradas foi possível perceber que o estado lógico de *Fgf8* não era sempre complementar ao de *Sp8*, fazendo com que as regras lógicas obtidas, de fato, não indicassem sempre a ativação. Esse problema é oriundo do método de discretização utilizado. Como será apresentado na Seção **4.6.3**, ao trocar o método de discretização tal relação passou a ser obtida em todas as execuções independentes da CGP.

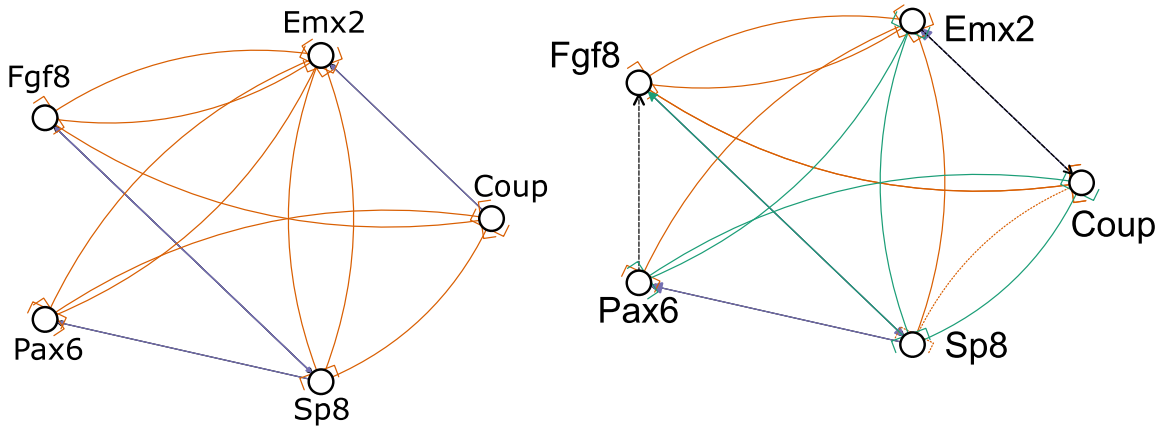
4.2.2 Problemas Experimentais

Para os problemas experimentais, o mesmo pré-processamento utilizado anteriormente foi aplicado. A discretização foi realizada utilizando o próprio módulo de discretização do *framework* CGPGRN com os método Bikmeans para todos os problemas. Aqui, considerou-se também a etapa de agrupamento utilizando K-Means e número máximo de *clusters* igual a 10. O melhor número de *clusters* é aquele que maximiza o coeficiente de silhueta.

Os parâmetros da CGP são variáveis em relação ao número de genes considerados.

Figura 51 – Rede mCAD reconstruída com 0% *dropout*. Linhas azuis representam ativação e laranjas, inibição. Linhas sólidas são relações regulatórias corretas, linhas tracejadas são aquelas obtidas somente pela proposta, e linhas verdes são relações presentes na rede *ground-truth* que a proposta não encontrou.

(a) Rede *Ground-Truth*. (b) Rede reconstruída.



Fonte: SILVA *et al.* (2023).

Quanto maior a quantidade de espécies envolvidas, mais recurso computacional deve disponibilizado para a CGP e maior deverá ser o tamanho da representação do indivíduo (n_c). Isso pode ser definido pelo usuário explicitamente ou pode-se deixar o cálculo automático do *framework*. Para este experimento, utilizou-se o cálculo automático. Para a evolução de um único circuito com múltiplas saídas, o número máximo de nós (n_c) é dado por:

$$n_c = \begin{cases} 500, & \text{se } N_{esp} < 200 \\ 500 + ceil(\lfloor N_{esp}/200 \rfloor) \times 500, & \text{caso contrário} \end{cases} \tag{4.1}$$

onde N_{esp} é o número de espécies envolvidas.

Já para o número de avaliações da função objetivo (n_{eval}), o cálculo é dado por:

$$n_{eval} = 1.500.000 + ceil(\lfloor N_{esp}/400 \rfloor) \times 1.500.000 \tag{4.2}$$

Ambas fórmulas foram derivadas empiricamente através de estudos preliminares.

O conjunto de funções é Γ = {AND, OR, NOT, NOR, XOR, NAND, XNOR}. Os circuitos foram evoluídos considerando todas as saídas simultaneamente, e não somente uma como no experimento anterior. Isso maximiza a eficiência da avaliação dos indivíduos em GPU, que agora também é considerada. Os demais parâmetros de representação n_r = 1 e lb = n_c são os mesmos do experimento anterior bem como o operador de mutação SAM. O número de execuções independentes foi aumentado para 10 a fim de refinar as probabilidades de ocorrência das relações regulatórias. Todos os problemas experimentais,

descritos na Seção 4.1, foram considerados. O número de espécies de cada problema para cada configuração é apresentado na Tabela 13

Tabela 13 – Número de espécies de cada problema em cada configuração.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
500nTF	500	500	500	500	500	500	500
500TF	910	948	821	1.120	704	632	560
1000nTF	1.000	1.000	1.000	1.000	1.000	1.000	692
1000TF	1.410	1.448	1.321	1.620	1.204	1.132	692

Fonte: Elaborado pelo autor (2024).

Os algoritmos utilizados para comparação são GENIE3, GRNBOOST2, PIDC, PPCOR e SINCERITIES, uma vez que foram apontados como os algoritmos que apresentaram melhores desempenhos em dados experimentais (PRATAPA *et al.*, 2020). Além disso, as três redes de referência (STRING, NonSpecific e ChIP-Seq) foram consideradas. É importante ressaltar que tais redes contém informações sobre possíveis relações regulatórias de todas as espécies presentes no conjunto de dados, conforme apresentado na Tabela 11. Para os subconjuntos das configurações de interesse (500nTF, 500TF, 1000nTF e 1000TF) as redes de referência são construídas intersectando as espécies presentes em cada uma das configurações com as redes de referência completas.

Mantendo a base de comparação apresentada em (PRATAPA *et al.*, 2020), a métrica de referência utilizada aqui é o EP. Um fato importante a ser levantado nesse momento é que o *framework* de avaliação BEELINE não foi capaz de calcular os valores de AUPRC e AUROC para os dados experimentais por conta do alto consumo de memória (superior a 16GB). Por outro lado, o módulo de avaliação do *framework* CGPGRN é capaz de lidar com essa grande quantidade de dados e as avaliações dos dados experimentais sob o ponto de vista de diversas métricas foi apresentado na Seção 2.13 e é discutido em detalhes na Seção 4.7. As Tabelas 14, 15, 16, e 17 apresentam os resultados de melhor caso para todos os problemas e todos os algoritmos, para as configurações 500nTF, 500TF, 1000nTF, e 1000TF, respectivamente. Os resultados tabulares sobre o caso da mediana estão disponíveis no material suplementar no repositório. Células com "-" indicam que o algoritmo não obteve nenhuma relação regulatória correta para problema (apresentado na primeira coluna) na dada rede (uma das linhas, por problema).

De acordo com a Tabela 14 é possível perceber que a CGP obteve melhores resultados em 9/21 situações. A maior parte deles (6) foi obtido quando considerada a rede de referência ChIP-Seq. O segundo melhor desempenho foi alcançado pelo PIDC, com melhores resultados em 7/21 situações, sendo 4 deles na rede de referência STRING. O GRNBOOST2 obteve melhores resultados em 2/21 situações e o GENIE3 e SINCERITIES

Tabela 14 – Resultados para todos os algoritmos e problemas considerando os melhores valores para a configuração 500nTF. Melhores resultados são apresentados em **negrito**.

Problem	Network	PIDC	GENIE3	GRNBOOST2	SINCERITIES	CGP
hESC	STRING	-	-	-	-	**0,0017**
	NonSpecific	-	-	-	-	-
	ChIP-Seq	-	0,0035	0,0088	0,0053	**0,0122**
hHep	STRING	0,0096	0,0024	0,0048	0,0024	**0,0097**
	NonSpecific	-	-	-	-	**0,0008**
	ChIP-Seq	-	-	-	-	**0,0112**
mDC	STRING	**0,0079**	-	-	-	-
	NonSpecific	-	-	-	-	-
	ChIP-Seq	-	-	-	-	**0,0079**
mESC	STRING	0,0278	**0,0478**	**0,0478**	0,0080	0,0033
	NonSpecific	-	0,0053	0,0053	**0,0106**	0,0049
	ChIP-Seq	0,0137	0,0265	**0,0298**	0,0125	0,0195
mHSC-E	STRING	**0,0694**	0,0451	0,0382	-	0,0024
	NonSpecific	**0,0435**	0,0254	0,0326	-	0,0042
	ChIP-Seq	0,0066	0,0036	0,0036	0,0058	**0,0110**
mHSC-L	STRING	**0,1724**	0,1043	0,0435	-	0,0016
	NonSpecific	**0,0550**	0,0275	0,0229	-	0,0037
	ChIP-Seq	0,0109	0,0118	0,0139	0,0066	**0,0148**
mHSC-GM	STRING	**0,1161**	0,0536	0,0536	-	0,0040
	NonSpecific	**0,0808**	0,0102	0,0254	-	0,0023
	ChIP-Seq	0,0122	0,0086	0,0108	0,0108	**0,0171**

Fonte: Elaborado pelo autor (2024).

em 1/21 situações cada, sendo que o único melhor resultado de GENIE3 está empatado com o resultado obtido pelo GRNBOOST2. Nenhum dos algoritmos foi capaz de encontrar relações regulatórias em todas as situações. Contudo, a CGP foi o método com menor quantidade de soluções não obtidas (3), seguido por GENIE3 e GRNBOOST2 com 7 e PIDC com 8. O algoritmo SINCERITIES não apresentou bons resultados para a maioria dos problemas e não foi capaz de encontrar relações regulatórias corretas em 13 situações.

Analisando a Tabela 15, que consiste na configuração 500TF é possível perceber que o PIDC foi o algoritmo com melhores resultados na maioria das situações (13/21 problemas). A CGP só foi capaz de obter melhores resultados em 3/21 problemas, todos eles quando considerada a rede ChIP-Seq. Os algoritmos GENIE3 e GRNBOOST2 ficaram ambos com 2/21 melhores resultados e, novamente, SINCERITIES obteve melhores resultados em 1/21 situações. A CGP não foi capaz de encontrar relações regulatórias corretas para o problema hESC quando considerada a rede NonSpecific. SINCERITIES apresentou o pior desempenho, não sendo capaz de encontrar relações regulatórias corretas em 9/21

Tabela 15 – Resultados para todos os algoritmos e problemas considerando os melhores valores para a configuração 500TF. Melhores resultados são apresentados em **negrito**.

Problem	Network	PIDC	GENIE3	GRNBOOST2	SINCERITIES	CGP
hESC	STRING	**0,0296**	0,0108	0,0176	-	0,0006
	NonSpecific	**0,0119**	0,0055	0,0058	-	-
	ChIP-Seq	0,0044	**0,0086**	0,0079	-	0,0058
hHep	STRING	**0,0373**	0,0175	0,0178	0,0088	0,0149
	NonSpecific	**0,0126**	0,0061	0,0068	0,0034	0,0063
	ChIP-Seq	0,0153	0,0071	0,0078	**0,0159**	0,0153
mDC	STRING	**0,0293**	0,0131	0,0125	0,0075	0,0278
	NonSpecific	**0,0228**	0,0098	0,0062	0,0026	0,0184
	ChIP-Seq	**0,0026**	0,0013	0,0013	-	0,0013
mESC	STRING	**0,0377**	0,0215	0,0251	0,0120	0,0060
	NonSpecific	**0,0191**	0,0139	0,0149	0,0074	0,0065
	ChIP-Seq	0,0304	0,0348	**0,0351**	0,0235	0,0290
mHSC-E	STRING	**0,0598**	0,0387	0,0401	-	0,0064
	NonSpecific	**0,0372**	0,0309	0,0302	0,0007	0,0048
	ChIP-Seq	0,0177	0,0030	0,0062	0,0151	**0,0253**
mHSC-L	STRING	**0,1522**	0,0949	0,0657	-	0,0033
	NonSpecific	**0,0607**	0,0287	0,0287	-	0,0022
	ChIP-Seq	0,0107	0,0130	0,0143	0,0068	**0,0172**
mHSC-GM	STRING	0,0709	**0,0722**	0,0709	-	0,0067
	NonSpecific	0,0404	0,0391	**0,0418**	-	0,0050
	ChIP-Seq	0,0137	0,0099	0,0151	0,0124	**0,0219**

Fonte: Elaborado pelo autor (2024).

situações. Os demais algoritmos (PIDC, GENIE3 e GRNBOOST2) encontraram soluções para todos os problemas em todas as redes de referência.

Quando considerados os resultados da configuração 1000nTF apresentados na Tabela 16, a CGP obteve os melhores resultados em 5/21 situações, sendo 3 delas na rede ChIP-Seq. PIDC obteve melhores resultados em 6/21 situações, sendo a maioria delas (3) na rede STRING. GRNBOOST2 também obteve melhores resultados em 6/21 situações. Por fim, GENIE3 obteve melhores resultados em 4/21 situações e SINCERITIES não obteve nenhum melhor resultado. Em relação ao número de vezes em que os algoritmos não conseguiram obter relações regulatórias corretas, SINCERITIES ocupa o primeiro lugar com 12 ocorrências, seguido por PIDC, GRNBOOST2 e CGP com 5 ocorrências e GENIE3 com 4 ocorrências.

Por fim, conforme apresentado na Tabela 17, PIDC apresenta os melhores resultados em 13/21 situações, seguido por CGP com 4/21 situações, GENIE3 com 3/21 situações, GRNBOOST2 com 2/21 situações e SINCERITIES não obteve melhores resultados em

Tabela 16 – Resultados para todos os algoritmos e problemas considerando os melhores valores para a configuração 1000nTF. Melhores resultados são apresentados em **negrito**.

Problem	Network	PIDC	GENIE3	GRNBOOST2	SINCERITIES	CGP
hESC	STRING	-	-	-	-	**0,0003**
	NonSpecific	-	-	-	-	**0,0016**
	ChIP-Seq	0,0038	**0,0070**	0,0052	-	0,0042
hHep	STRING	0,0054	0,0045	**0,0063**	0,0009	0,0023
	NonSpecific	-	-	-	-	-
	ChIP-Seq	0,0014	0,0005	**0,0019**	0,0009	-
mDC	STRING	**0,0071**	0,0012	0,0024	-	0,0007
	NonSpecific	-	**0,0020**	-	-	0,0011
	ChIP-Seq	**0,0032**	-	-	-	-
mESC	STRING	0,0071	**0,0297**	**0,0297**	0,0014	0,0021
	NonSpecific	-	0,0086	**0,0095**	0,0026	0,0019
	ChIP-Seq	0,0170	**0,0262**	0,0257	0,0107	0,0137
mHSC-E	STRING	**0,0512**	0,0216	0,0243	0,0014	0,0021
	NonSpecific	**0,0249**	0,0132	0,0190	-	0,0011
	ChIP-Seq	0,0046	0,0010	0,0015	0,0050	**0,0083**
mHSC-L	STRING	**0,1039**	0,0584	0,0455	-	-
	NonSpecific	**0,0346**	0,0221	0,0158	-	-
	ChIP-Seq	0,0097	0,0095	0,0100	0,0046	**0,0114**
mHSC-GM	STRING	0,0273	0,0375	**0,0444**	-	0,0014
	NonSpecific	0,0194	0,0153	**0,0209**	-	0,0017
	ChIP-Seq	0,0080	0,0054	0,0070	0,0069	**0,0134**

Fonte: Elaborado pelo autor (2024).

nenhuma das situações. Em relação as situações onde não foram encontradas relações regulatórias corretas, SINCERITIES novamente apresenta o pior desempenho, não encontrando relações regulatórias corretas em 8 situações, seguido por CGP (2) e GRNBOOST2 (1). PIDC e GENIE3 encontraram ao menos uma relação regulatória correta para todos os problemas quando consideradas todas as redes de referência.

A Tabela 18 apresenta a contagem de desempenho de todos os algoritmos, em todos os problemas considerando as três redes de referência, contendo informações sobre quantas vezes cada método encontrou o melhor resultado (#MR) e quantas vezes cada método não foi capaz de obter relações regulatórias (#SS). A partir dela é possível concluir que o PIDC obteve melhores resultados 39 vezes, sendo o algoritmo com melhores resultados quando consideradas as redes de referência STRING e NonSpecific. Além disso, o PIDC tende a apresentar bons resultados nas configurações que levam em consideração a presença dos fatores de transcrição. A CGP obteve os melhores resultados somente para a rede ChIP-Seq. SINCERITIES é o algoritmo com pior desempenho, não sendo capaz de encontrar relações

Tabela 17 – Resultados para todos os algoritmos e problemas considerando os melhores valores para a configuração 1000TF. Melhores resultados são apresentados em **negrito**.

Problem	Network	PIDC	GENIE3	GRNBOOST2	SINCERITIES	CGP
hESC	STRING	**0,0188**	0,0099	0,0146	0,0041	0,0044
	NonSpecific	**0,0087**	0,0041	0,0050	0,0080	0,0026
	ChIP-Seq	0,0032	**0,0085**	0,0066	0,0079	0,0031
hHep	STRING	**0,0238**	0,0128	0,0129	0,0037	0,0098
	NonSpecific	**0,0093**	0,0052	0,0052	0,0019	0,0078
	ChIP-Seq	**0,0118**	0,0051	0,0063	0,0110	**0,0118**
mDC	STRING	**0,0220**	0,0098	0,0103	0,0027	0,0044
	NonSpecific	**0,0181**	0,0074	0,0064	0,0013	0,0029
	ChIP-Seq	**0,0017**	0,0008	-	-	-
mESC	STRING	**0,0296**	0,0175	0,0171	0,0061	0,0055
	NonSpecific	**0,0158**	0,0101	0,0115	0,0045	0,0044
	ChIP-Seq	0,0224	0,0271	**0,0281**	0,0176	0,0176
mHSC-E	STRING	**0,0389**	0,0279	0,0318	-	0,0021
	NonSpecific	0,0184	**0,0235**	0,0224	-	0,0024
	ChIP-Seq	0,0126	0,0023	0,0051	-	**0,0149**
mHSC-L	STRING	**0,1039**	0,0584	0,0455	-	-
	NonSpecific	**0,0346**	0,0221	0,0189	-	0,0019
	ChIP-Seq	0,0097	0,0097	0,0116	0,0046	**0,0183**
mHSC-GM	STRING	0,0267	**0,0420**	0,0404	-	0,0025
	NonSpecific	0,0206	0,0243	**0,0258**	-	0,0019
	ChIP-Seq	0,0083	0,0060	0,0082	0,0080	**0,0123**

Fonte: Elaborado pelo autor (2024).

regulatórias corretas em 42 situações, seguido por PIDC com 13 e os demais algoritmos ficaram empatados com 11 ocorrências. A explicação mais plausível para o desempenho da CGP não ser o melhor em todos os casos reside no fato de que o *framework* BEELINE leva em consideração somente as top-k relações regulatórias, sendo k a quantidade de relações regulatórias presentes na rede de referência reduzida (com as espécies alvo intersectadas nas redes completas). Os melhores resultados obtidos pela CGP estão concentradas na rede de referência ChIP-Seq que é, em todos os casos, a rede de referência com maior quantidade de relações regulatórias. Testes foram realizados considerando a rede completa e, em todos os casos, a CGP apresenta melhores resultados. Nessa situação, todas as relações regulatórias encontradas pela CGP são levadas em consideração na hora de computar a EP. Um outro fator que pode auxiliar a CGP a refinar suas soluções é aumentar a quantidade de execuções de rede para geração das probabilidades. No caso do experimento atual, uma relação regulatória encontrada em 1/10 execuções aparece com 10% de probabilidade de ocorrência. Se forem realizadas 20 execuções, essa probabilidade de ocorrência, caso

mantenha-se acontecendo uma única vez, cairia para 5% e, possivelmente, seria jogada para o final da lista ordenada de relações regulatórias. Como leva-se em consideração somente as top-k relações regulatórias, essa ocorrência de 5% poderia ser eliminada da contagem no momento da avaliação. Outro fator importante a ser levantado é o uso da etapa de agrupamento. Essa etapa mostrou-se favorável e auxiliou a CGP a encontrar soluções factíveis em estudos realizados previamente. Contudo, como será discutido na Seção 4.7, a geração dos conjuntos de dados apresentados em (PRATAPA *et al.*, 2020) reúne genes que nunca compartilham relações regulatórias, o que prejudica o agrupamento.

Tabela 18 – Contagem de desempenho dos algoritmos para todos os problemas. #MR indica o número de vezes que o método obteve melhores resultados e #SS indica o número de vezes em que o método não encontrou relações regulatórias. Os melhores resultados de #MR por rede de referência estão apresentados em **negrito**.

	PIDC		GENIE3		GRNBOOST2		SINCERITIES		CGP	
	#MR	#SS	#MR	#SS	#MR	#SS	#MR	#SS	#MR	#SS
STRING	**19**	2	4	3	4	3	0	16	3	3
NonSpecific	**16**	8	2	5	4	5	1	18	2	5
ChIP-Seq	4	3	4	3	4	3	1	8	**16**	3
Totais	39	13	10	11	12	11	2	42	21	11

Fonte: Elaborado pelo autor (2024).

4.3 AVALIAÇÃO DO MÉTODO PROPOSTO EM DADOS DE ORGANISMOS AMPLAMENTE ESTUDADOS

Experimentos computacionais foram realizados a fim de avaliar o desempenho do método proposto em dados de organismos amplamente estudados na literatura e que também constituem conjuntos de dados utilizados com frequência para a validação de métodos de inferência de GRNs. Para essas avaliações, são considerados todos os problemas do Grupo 3 de conjuntos de dados apresentados na Seção 4.1.

Primeiramente, considerando o problema da rede DNA SOS da *E. coli*, para CGP foram adotadas 50.000 avaliações da função objetivo, $n_r = 1$, $n_c = \text{lb} = 100$, 5 execuções independentes e $\Gamma = \{\text{AND, OR, NOT, NOR, XOR, NAND, XNOR}\}$. Os resultados para as 4 redes considerando AUPRC e AUROC de todos os algoritmos são apresentados na Tabela 19 e nos *boxplots* da Figura 52.

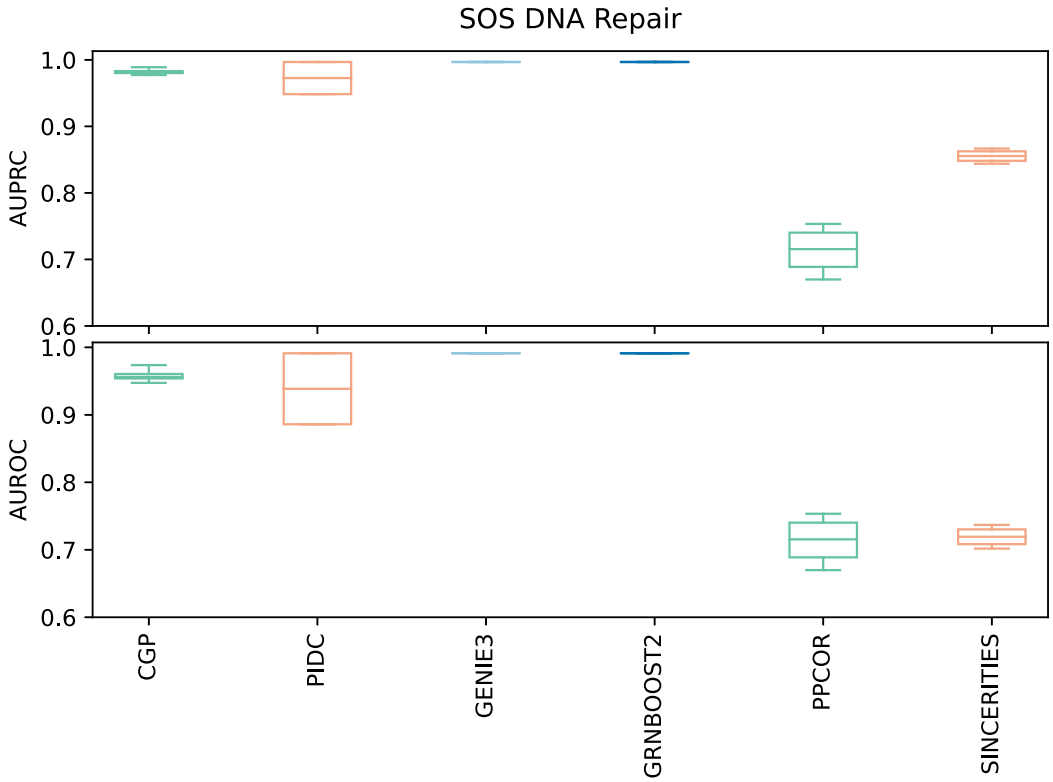
De acordo com os resultados apresentados na Tabela 19 é possível perceber que os melhores resultados, tanto para AUPRC quanto para AUROC, em todos os casos, foi obtido pelos algoritmos GENIE3 e GRNBOOST2. O resultado ser o mesmo é justificado pelo fato de que ambos os algoritmos utilizam o mesmo princípio de funcionamento, conforme discutido anteriormente. A diferença reside na adequação do GRNBOOST2 para *datasets*

Tabela 19 – Resultados de AUPRC e AUROC para todas as redes SOS para todos os algoritmos. Melhores valores são apresentados em negrito.

Alg.	SOS-1		SOS-2		SOS-3		SOS-4	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
CGP	0,9928	0,9824	0,9889	0,9737	0,9811	0,9561	0,9889	0,9737
PIDC	0,9484	0,8860	0,9484	0,8860	**0,9968**	**0,9912**	**0,9968**	**0,9912**
GENIE3	**0,9968**	**0,9912**	**0,9968**	**0,9912**	**0,9968**	**0,9912**	**0,9968**	**0,9912**
GRNBOOST2	**0,9968**	**0,9912**	**0,9968**	**0,9912**	**0,9968**	**0,9912**	**0,9968**	**0,9912**
PPCOR	0,7534	0,5877	0,6951	0,5351	0,6699	0,5175	0,7359	0,5702
SINCERITIES	0,8497	0,7105	0,8611	0,7281	0,8666	0,7368	0,8439	0,7018

maiores em termos de tempo computacional. PIDC foi capaz de igualar esses resultados para as redes 3 e 4. A CGP, apesar de não encontrar os melhores resultados, sempre obteve valores próximos a 0,99 em AUPRC e sempre superiores a 0,95 para AUROC, em todos os casos. Esses resultados foram superiores aos obtidos pelo PIDC nas duas primeiras redes e superior a todos os resultados obtidos pelo PPCOR e SINCERITIES. Além disso, PPCOR apresenta os piores resultados nestes problemas.

Figura 52 – *Boxplots* de AUPRC e AUROC para todas as redes SOS e todos os algoritmos.



Fonte: Elaborado pelo autor (2024).

Já para os *boxplots* apresentados na Figura 52, GENIE3 e GRNBOOST2, além de obter os mesmos valores para todos os problemas quando comparados entre si, também obtiveram os mesmos valores para todas as redes consideradas. Por esse motivo, os *boxplots* não apresentam variação. A CGP apresenta uma pequena variação em AUPRC, mantendo uma mediana em 0,9889 e uma mediana de 0,9737 em AUROC. PIDC apresentou maior

variabilidade entre as redes consideradas. PPCOR e SINCERITIES também apresentam uma variabilidade mas os resultados são sempre abaixo de 0,9 e 0,8 para AUPRC e abaixo de 0,8 para AUROC, para PPCOR e SINCERITIES, respectivamente.

De maneira geral, os resultados indicam que GENIE3, GRNBOOST2, CGP e PIDC foram capazes de reconstruir corretamente quase completamente as redes consideradas nestes experimentos.

Na sequência, os experimentos foram realizados considerando o conjunto de dados IRMA. Os parâmetros da CGP são os mesmos para os problemas DNA SOS. Os resultados de AUPRC e AUROC para todos os algoritmos são apresentados na Tabela 20.

Tabela 20 – Resultados de AUPRC e AUROC para todos os algoritmos considerando o problema IRMA. Melhores resultados são apresentados em negrito.

Metric	CGPGRN	GENIE3	GRNBOOST2	PIDC	PPCOR	SINCERITIES
AUPRC	**0,7466**	0,5826	0,5056	0,6911	0,4000	0,4351
AUROC	**0,7448**	0,6146	0,5313	0,6875	0,5000	0,5781

Os resultados indicam que o CGPGRN fornece os melhores resultados para o problema IRMA, tanto em AUPRC quanto em AUROC. PIDC apresentou resultados superiores aos obtidos pelo GENIE3 e GRNBOOST2. Novamente, PPCOR e SINCERITIES não apresentaram um bom desempenho.

Por fim, são considerados os problemas da competição DREAM. Para os problemas de 10 genes, os parâmetros da CGP são os mesmos dos experimentos anteriores. Contudo, para 100 genes, $n_c = lb = 1000$ e o número máximo de avaliações é 500.000, mantendo uma proporção de 10 vezes em relação ao aumento da quantidade de genes. Os demais parâmetros são os mesmos. As Tabelas 21 e 22 apresentam os resultados para todos os algoritmos e problemas, para AUPRC e AUROC, respectivamente.

De acordo com a Tabela 21 é possível perceber que, tanto para 10 quanto para 100 genes, o CGPGRN obteve os melhores resultados em todos os conjuntos de dados testados. Conforme apresentado em experimentos anteriores, PPCOR e SINCERITIES não apresentaram bom desempenho. Além disso, SINCERITIES não foi capaz de obter relações regulatórias corretas para nenhum dos problemas considerados aqui na categoria DREAM4-100.

Já para os valores de AUROC, de acordo com a Tabela 22, para os problemas da categoria DREAM4-10, o CGPGRN apresenta os melhores resultados em todos os casos. Contudo, para o DREAM4-100, GRNBOOST2 apresenta os melhores resultados para os 3 primeiros problemas e os dois últimos são melhores quando considerado o GENIE3. Novamente, PPCOR e SINCERITIES não apresentaram bom desempenho e SINCERITIES não foi capaz de encontrar relações regulatórias corretas para a categoria DREAM4-100. Uma possível explicação para os resultados inferiores obtidos pelo CGPGRN em AUROC

Tabela 21 – Resultados de AUPRC para todos os conjuntos de dados e todos os algoritmos. Melhores resultados estão em negrito

Alg.	#1	#2	#3	#4	#5
DREAM4-10					
CGPGRN	**0,3500**	**0,2833**	**0,3545**	**0,2887**	**0,3037**
GENIE3	0,2002	0,1762	0,1886	0,1974	0,1607
GRNBOOST2	0,1876	0,1725	0,1885	0,2154	0,1233
PIDC	0,1667	0,1345	0,1537	0,1427	0,0960
PPCOR	0,1667	0,1778	0,1537	0,2345	0,1336
SINCERITIES	0,1556	0,1577	0,2580	0,1297	0,1401
DREAM4-100					
CGPGRN	**0,0484**	**0,0535**	**0,0532**	**0,0321**	**0,0548**
GENIE3	0,0206	0,0352	0,0288	0,0239	0,0237
GRNBOOST2	0,0234	0,0349	0,0257	0,0251	0,0228
PIDC	0,0360	0,0267	0,0206	0,0204	0,0182
PPCOR	0,0178	0,0252	0,0197	0,0213	0,0195
SINCERITIES	-	-	-	-	-

Tabela 22 – Resultados de AUROC para todos os conjuntos de dados e todos os algoritmos. Melhores resultados estão em negrito

Alg.	#1	#2	#3	#4	#5
DREAM4-10					
CGPGRN	**0,6391**	**0,5836**	**0,6573**	**0,6129**	**0,5775**
GENIE3	0,5502	0,5144	0,5591	0,5375	0,5748
GRNBOOST2	0,5653	0,4654	0,5884	0,6014	0,4423
PIDC	0,5000	0,3809	0,4867	0,5040	0,3611
PPCOR	0,5000	0,5000	0,4867	0,5639	0,5075
SINCERITIES	0,4982	0,4586	0,5822	0,4026	0,5454
DREAM4-100					
CGPGRN	0,5121	0,5024	0,5092	0,5113	0,5121
GENIE3	0,5337	0,5943	0,5606	**0,5294**	**0,5433**
GRNBOOST2	**0,5735**	**0,5980**	**0,5324**	0,5269	0,5142
PIDC	0,5276	0,5157	0,5083	0,4926	0,4878
PPCOR	0,5000	0,5000	0,5000	0,5000	0,5000
SINCERITIES	-	-	-	-	-

para a categoria DREAM4-100 é a grande presença de falsos positivos. Dessa forma, a curva ROC é afetada. Uma alternativa para tentar contornar isso seria um refinamento melhor das soluções obtidas pelo CGPGRN, aumentando, por exemplo, o número de redes internas evoluídas por execução. Com isso, as probabilidades seriam geradas com base em um número maior de execuções.

4.4 COMPARAÇÃO DO MÉTODO PROPOSTO COM MÉTODOS BOOLEANOS E BASEADOS EM METAHEURÍSTICA

Conforme discutido na Seção 2.11, ainda que, exceto pelo algoritmo SCNS o conjunto fornecido por PRATAPA *et al.* (2020) não contemple métodos Booleanos e baseados em computação evolucionista, experimentos computacionais foram realizados para comparação entre o método proposto e o algoritmo ATEN, que contempla tanto a modelagem Booleana quanto o uso de *Simulated Annealing*, sendo uma abordagem mais próxima do método proposto. Além disso, dentre os métodos apresentados em PUŠNIK *et al.* (2022), GABNI, MIBNI, Best-Fit Extension e ATEN alcançaram resultados comparáveis em termos de precisão mas nenhum superou os demais e não há um vencedor claro. Como será discutido adiante, por conta da incapacidade em lidar com problemas que envolvam grandes quantidades de genes e o tempo computacional associado à execução do algoritmo, utilizamos os resultados apresentados em (PUŠNIK *et al.*, 2022) em termos de MCC e realizamos uma normalização em relação ao tempo computacional apresentado no trabalho, para todos os algoritmos (Seção 2.11). O código fonte do algoritmo ATEN é disponibilizado. Por esse motivo, 10 execuções independentes deste algoritmo foram realizadas em nosso ambiente computacional para cada quantidade de genes (16, 32 e 64). Para cada um desses casos, um valor de mediana é obtido. Dividindo cada um desses valores de mediana obtidos pelos valores apresentados no algoritmo ATEN em (PUŠNIK *et al.*, 2022), obtém-se 1,02, 1,03, e 1,04. Dessa forma, utilizamos a mediana desses valores (1,03) para realizar a normalização dos tempos apresentados em (PUŠNIK *et al.*, 2022) para os demais algoritmos. Contudo, é importante ressaltar que apenas GABNI e ATEN são paralelizados.

Os problemas considerados são os mesmos apresentados em (PUŠNIK *et al.*, 2022) para os dados sintéticos de *E. coli* contendo 16, 32 e 64 genes e brevemente discutidos na Seção 4.1. Para a CGP foram consideradas 5 execuções independentes, 50.000 avaliações da função objetivo e $\Gamma = \{\text{AND, OR, NOT, NOR, XOR, NAND, XNOR}\}$.

Os resultados são apresentados na Tabela 23. A partir da tabela é possível perceber que para 16 e 32 genes a CGP obteve o melhor resultado em 8/10 redes consideradas. Já para 64 genes, a CGP obteve melhores resultados em 9/10 redes. No total, a CGP obteve melhores resultados em 25/30 problemas considerados. Em algumas redes, como por exemplo a #7, os resultados obtidos pela CGP são significativamente maiores que os obtidos pelo ATEN, chegando a 235%, 413% e 255% para 16, 32 e 64 genes, respectivamente, com referência ao ATEN. Já para os casos em que a CGP obtém piores resultados, a diferença relativa fica entre 73,3% e 89,7% para as redes 2 e 8, considerando 16 genes, entre 56,4% e 71,6% para as redes 4 e 3, considerando 32 genes e de 13,6% para a rede 8 de 64 genes.

Os tempos computacionais normalizados em relação ao ambiente computacional no qual o método proposto é executado, em paralelo em GPU, com processador AMD®

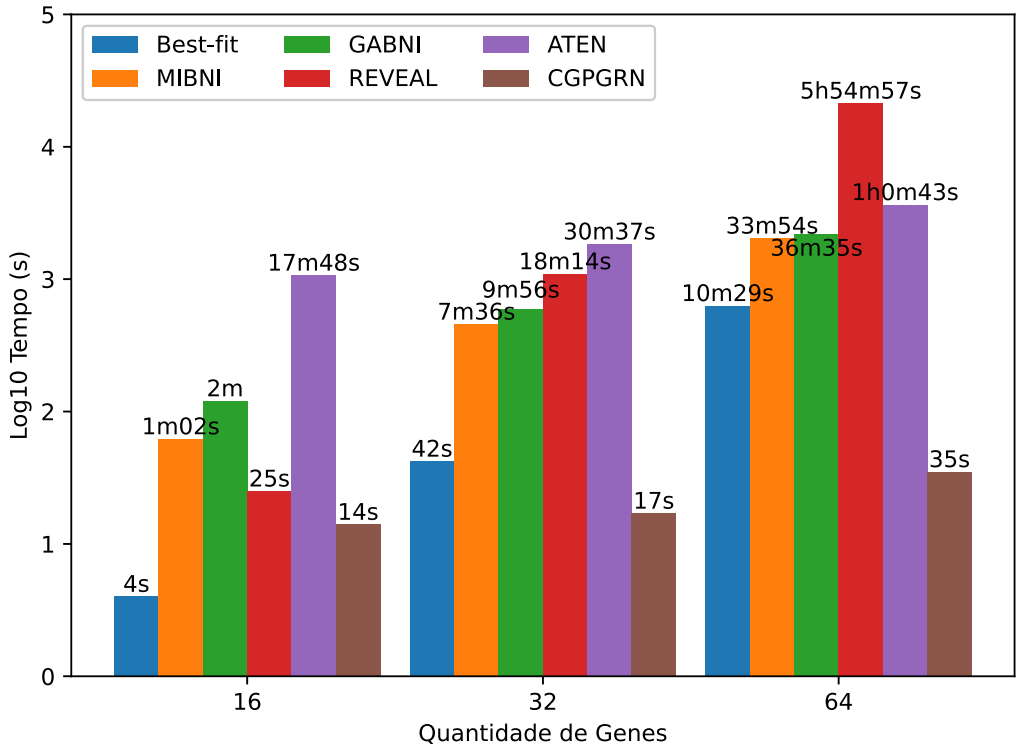
Tabela 23 – Resultados comparativos em relação ao MCC entre ATEN e CGP para todas as configurações de genes em todas as redes. Melhores valores estão em negrito.

Alg	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
					16 genes					
ATEN	0,0904	**0,1075**	0,0250	-0,0608	-0,0216	-0,0349	0,0716	**0,0728**	-0,1241	0,0723
CGP	**0,1557**	0,0287	**0,1553**	**0,1113**	**0,1333**	**0,0705**	**0,2397**	0,0075	**0,1945**	**0,0938**
					32 genes					
ATEN	0,0157	0,0415	**0,0726**	**0,0126**	-0,0135	-0,0200	0,0090	-0,0049	0,0044	-0,0508
CGP	**0,0635**	**0,0619**	0,0206	0,0055	**0,0507**	**0,0958**	**0,0462**	**0,0375**	**0,0769**	**0,0293**
					64 genes					
ATEN	0,0187	0,0128	-0,018	0,0214	0,0026	0,0023	0,0087	**0,0213**	0,0308	0,0003
CGP	**0,0321**	**0,0247**	**0,0196**	**0,0563**	**0,0210**	**0,0213**	**0,0309**	0,0184	**0,0382**	**0,0342**

Ryzen®7 5800X, 64GB RAM DDR4 e GPU RTX3060 com 12GB GDDR5, são apresentados na Figura 53. O eixo das ordenadas é apresentado em escala logarítmica e os tempos médios de execução para cada algoritmo são apresentados no topo das barras. A partir das barras é possível perceber que o *Best-fit* apresenta o menor tempo computacional para 16 genes, seguido pelo CGPGRN e o REVEAL. Para 32 e 64 genes, o CGPGRN apresenta o menor tempo computacional. De acordo com PUŠNIK *et al.* (2022), o algoritmo ATEN apresenta a melhor escalabilidade dentre os métodos discutidos. Contudo, é possível perceber que o CGPGRN apresenta escalabilidade melhor que todos os algoritmos comparados, exibindo um crescimento linear com o aumento exponencial de entradas. Outro fato importante a ser destacado é que o número de linhas das tabelas verdade (número de combinações em relação ao número de entradas) não é necessariamente exponencial, uma vez que não existe garantia de que todas as combinações estejam presentes no conjunto de dados discretizados. Isso é justificado pelo fato de que nem todos os estados discretos podem ter sido observados na discretização da amostra analisada.

Além disso, é importante ressaltar as limitações de todos os algoritmos comparados, conforme apresentado em (PUŠNIK *et al.*, 2022). Os algoritmos são capazes de lidar com no máximo 150 genes e com uma quantidade máxima de reguladores de 60 genes. A explicação para tal fato é justamente os problemas de escalabilidade de métodos Booleanos associados ao crescimento exponencial em relação ao número de entradas. Entretanto, o CGPGRN não possui limitação de número máximo de genes, conforme demonstrado em experimentos anteriores com até 1620 genes (Tabela 13), não possui limitação de número de reguladores (assume-se que todo e qualquer gene pode compartilhar relações regulatórias), apresentou os menores tempos computacionais dentre os métodos de inferência de GRNs Booleanas e apresentou os melhores resultados em 25/30 problemas testados em relação ao método Booleano mais recente ATEN (2020). Ainda, tendo em vista o uso de *Simulated Annealing* no ATEN, os resultados do CGPGRN também indicam o uso de uma melhor metaheurística no contexto de inferência de GRNs.

Figura 53 – Tempos computacionais para a inferência de GRNs considerando todos os algoritmos e todas as quantidades de genes.



Fonte: Elaborado pelo autor (2024).

4.5 AVALIAÇÃO DO MÉTODO PROPOSTO PARA MODELOS CONTÍNUOS

Para avaliar o método proposto, foram utilizados os dados do ritmo circadiano da *Drosophila* com 5 e 10 espécies (GOLDBETER, 1995; LELOUP; GOLDBETER, 1998), conforme descrito na Seção 4.1. Como estes dados são resultantes de simulação, a etapa de pré-processamento não é necessária, uma vez que não existem *dropouts*. Como parâmetros da CGP, adotou-se 20 execuções independentes, 50.000 avaliações da função objetivo, $n_r = 1$, $n_c = 100$, lb $= n_c$ e o conjunto de funções $\Gamma = \{\text{AND, OR, NOT, XOR}\}$. Estes parâmetros foram determinados empiricamente em testes preliminares.

Para a determinação dos coeficientes numéricos τ, z e k, uma $(\mu + \lambda)$-ES, sem recombinação e com a mutação apresentada em (BEYER; SCHWEFEL, 2002) foi utilizada. Segundo essa ES, dado um indivíduo $p \in P^{(t)} = (x, \Psi)$, seus parâmetros são mutados como segue:

$$\begin{aligned} \tau &= \frac{c}{\sqrt{2\times\sqrt{size}}} \\ \tau_0 &= \frac{c}{\sqrt{2\times size}} \\ \Psi_n &= \Psi_{n-1} \times e^{(\tau_0\times N(0,1))+(\tau\times N(0,1))} \\ x_n &= x_{n-1} \times N(0,1) \end{aligned} \tag{4.3}$$

onde *size* é o tamanho do indivíduo (neste trabalho, a quantidade de coeficientes), n é a

geração atual, *N(0,1)* é uma distribuição normal de média zero e desvio padrão 1, e c é o parâmetro de aprendizado.

Foram adotados os parâmetros $\mu = 15$ e $\lambda = 105$, já que tipicamente $\lambda = 7 \times \mu$, e um número máximo de avaliações da função objetivo igual a 10.000. Para obter um modelo simplificado, somente a solução com menor número de elementos lógicos obtidos na CGP foi utilizada para a obtenção do modelo contínuo. O código fonte está disponível[6]. Os métodos foram implementados utilizando C++ e o integrador numérico LSODA[7], que resolve numericamente o problema de valor inicial do sistema de equações diferenciais de primeira ordem. Além disso, $z \in \mathbb{Z}$ e $k, \tau \in \mathbb{R}$, e estas variáveis são limitadas a $1 \leq z \leq 25$, $0.1 \leq k \leq 1$ e $0.1 \leq \tau \leq 5$ (KRUMSIEK; WITTMANN; THEIS, 2011). Na ES, as variáveis são representadas e evoluídas em $\mathbb{R}$ e somente a parte inteira (truncamento) é usada para os valores de z.

É importante destacar que os BooleCubes e os HillCubes utilizam dados normalizados e existem n_g coeficientes z, k e τ, onde n_g é o número de genes envolvidos na rede, conforme descrito na Seção **2.11.2**. Aqui a normalização é realizada dentro do sistema de EDOs durante a integração numérica. Um exemplo dessa normalização e dos diferentes coeficientes são apresentados na Equação 4.4.

$$\dot{\overline{G2}} = \frac{1}{\tau_{G2}} \left(\frac{N(\overline{G1})^{z_{G2G1}}}{N(\overline{G1})^{z_{G2G1}} + k_{G2G1}^{z_{G2G1}}} - N(\overline{G2}) \right), \tag{4.4}$$

onde $N(v) = \frac{v}{\max(v)}$, $G1$ e $G2$ são variáveis de interesse, e $\max(G1)$ e $\max(G2)$ são, respectivamente, seus valores máximos nos dados. Os parâmetros z e k estão subescritos com G2G1, que significa a influência de G1 em G2.

Além disso, para a etapa de discretização, foi considerado o TSD (*transitional state discrimination*), apresentado na Seção **2.11.1**.

Para comparar os resultados numéricos das EDOs, utiliza-se a minimização da soma das diferenças absolutas entre os valores calculados pela integração numérica e os dos dados originais (norma 1). Experimentos preliminares foram realizados com a minimização da diferença quadrática (norma 2), mas os resultados foram piores do que os obtidos com o uso da norma 1, devido ao fato de que os valores numéricos eram muito pequenos e a norma 2 facilitava a introdução de erros numéricos (SILVA *et al.*, 2020). Além disso, é importante destacar que neste trabalho, o modelo candidato é integrado a partir de um valor inicial conhecido. Isso gera um conjunto de valores aproximados. Os dados originais são normalizados no intervalo [0, 1], conforme discutido na Seção 3.5. Por fim, os valores aproximados são comparados com esses valores esperados normalizados. O resultado é a soma em módulo das diferenças entre os valores esperados e previstos.

[6] `https://github.com/ciml/`
[7] `https://github.com/dilawar/libsoda-cxx`

Para fins de comparação dos resultados obtidos pela estratégia adotada, comparam-se os resultados com aqueles obtidos por uma regressão simbólica via Programação Genética através da biblioteca DEAP[8] para Python. Como primitivas, foram adotadas as operações aritméticas básicas (adição, subtração, multiplicação e divisão protegida), potência e constante inteira 0 ou 1. Em todos os casos, foram adotados 300 indivíduos na população, 10.000 avaliações da função objetivo, seleção por torneio de tamanho 3, recombinação de um ponto com 50% de probabilidade e mutação uniforme com 10% de probabilidade. A função de aptidão é a soma em módulo da diferença absoluta entre os valores obtidos pelo modelo e os valores reais.

Os resultados apresentados nesta seção são refinamentos da metodologia do método que propusemos em (SILVA *et al.*, 2020), contemplando a divisão dos dados em treino e teste. Os resultados originais considerando o conjunto de dados completo também está disponível no material suplementar no repositório.

4.5.1 Ritmo Circadiano com 5 espécies

O primeiro modelo criado é para o ritmo circadiano com 5 espécies, onde cada uma das espécies é nomeada aqui como A, B, C, D e E, representando M, P0, P1, P2 e PN, respectivamente, do modelo esquemático apresentado na Figura 45 na Seção 4.1. Os dados na forma de séries temporais são divididos em conjuntos de treino e teste, contemplando 70% e 30% do total dos dados, respectivamente. Os dados de treino discretizados e o diagrama de transição de estados são apresentados na Figura 54a, 54b e 54c, respectivamente. Os valores 0 e 1 foram convertidos em relação ao eixo das ordenadas para melhorar a legibilidade. Dez estados diferentes e 11 transições de estado foram determinados. Neste experimento não houve ambiguidades nas transições de estados. Espera-se uma tabela verdade composta de $2^5 = 32$ linhas, pois o fenômeno modelado envolve 5 variáveis. Conforme indicado na Seção 3.3, os estados não observados são considerados estados de irrelevância, e são apresentados na Tabela 24.

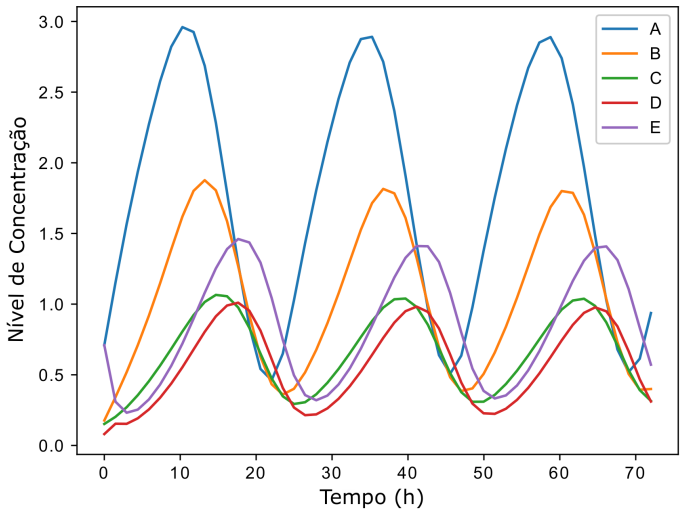
A CGP é aplicada à tabela verdade apresentada na Tabela 24 para obter o modelo discreto. A solução mais simples em termos de número de elementos lógicos foi selecionado como o modelo booleano para ser transformado em modelo contínuo. As expressões lógicas desse modelo são apresentadas na Tabela 25. Essas expressões lógicas foram utilizadas para determinar os BooleCubes e posteriormente convertidos em HillCubes. Os HillCubes são aplicados na Equação 2.40 e obtém-se o comportamento temporal das funções de atualização contínuas, apresentadas nas Equações 4.5, 4.6, 4.7, 4.8 e 4.9 para as variáveis A,B,C,D e E, respectivamente.

Por fim, os coeficientes numéricos concluem o modelo. Estes coeficientes, obtidos pela ES, e os gráficos dos dados reais e preditos pelo modelo são apresentados na Figura 55.

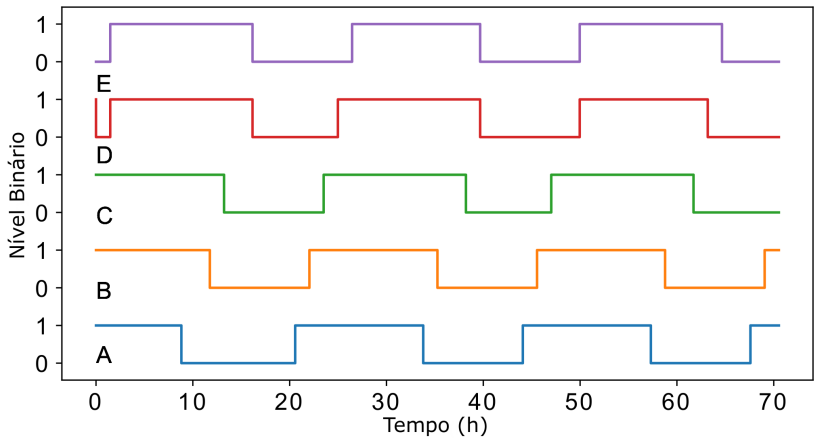
[8] `https://deap.readthedocs.io/en/master/`

Figura 54 – Gráficos e diagrama de transição de estados para o problema do ritmo circadiano de 5 variáveis.

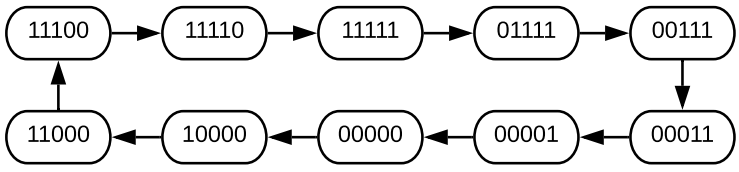
(a) Dados originais.



(b) Dados discretizados.



(c) Diagrama de transição de estados.



Fonte: SILVA *et al.* (2020).

Dentre as 20 execuções independentes, esse modelo alcançou um erro mínimo de 27,96. Uma possível explicação para o desempenho inferior observado para a variável B é o fato desta depender de E, sendo a única equação diferencial que representa uma relação lógica *AND*.

Tabela 24 – Tabela verdade completa para o ritmo circadiano de 5 variáveis.

t					t+1				
A	B	C	D	E	A	B	C	D	E
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	X	X	X	X	X
0	0	0	1	1	0	0	0	0	1
0	0	1	0	0	X	X	X	X	X
0	0	1	0	1	X	X	X	X	X
0	0	1	1	0	X	X	X	X	X
0	0	1	1	1	0	0	0	1	1
0	1	0	0	0	X	X	X	X	X
0	1	0	0	1	X	X	X	X	X
0	1	0	1	0	X	X	X	X	X
0	1	0	1	1	X	X	X	X	X
0	1	1	0	0	X	X	X	X	X
0	1	1	0	1	X	X	X	X	X
0	1	1	1	0	X	X	X	X	X
0	1	1	1	1	0	0	1	1	1
1	0	0	0	0	1	1	0	0	0
1	0	0	0	1	X	X	X	X	X
1	0	0	1	0	X	X	X	X	X
1	0	0	1	1	X	X	X	X	X
1	0	1	0	0	X	X	X	X	X
1	0	1	0	1	X	X	X	X	X
1	0	1	1	0	X	X	X	X	X
1	0	1	1	1	X	X	X	X	X
1	1	0	0	0	1	1	1	0	0
1	1	0	0	1	X	X	X	X	X
1	1	0	1	0	X	X	X	X	X
1	1	0	1	1	X	X	X	X	X
1	1	1	0	0	1	1	1	1	0
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1

Fonte: Elaborado pelo autor (2022).

Tabela 25 – Expressões lógicas para o ritmo circadiano de 5 variáveis.

Variável	Expressão
A	not(E)
B	not(E)
C	B + C
D	B + C
E	C × D

Fonte: SILVA *et al.* (2020).

$$\frac{dA}{dt} = \left(1 - \frac{N(E)^{n_{AE}}}{(N(E)^{n_{AE}} + k_{AE}^{n_{AE}})} - N(A)\right)/\tau_A \tag{4.5}$$

$$\frac{dB}{dt} = \left(1 - \frac{N(E)^{n_{BE}}}{(N(E)^{n_{BE}} + k_{BE}^{n_{BE}})} - N(B)\right)/\tau_B \tag{4.6}$$

$$\frac{dC}{dt} = \left(\left(\frac{N(B)^{n_{CB}}}{(N(B)^{n_{CB}} + k_{CB}^{n_{CB}})} + \frac{N(C)^{n_{CC}}}{(N(C)^{n_{CC}} + k_{CC}^{n_{CC}})}\right) - N(C)\right)/\tau_C \tag{4.7}$$

$$\frac{dD}{dt} = \left(\left(\frac{N(B)^{n_{DB}}}{(N(B)^{n_{DB}} + k_{DB}^{n_{DB}})} + \frac{N(C)^{n_{DC}}}{(N(C)^{n_{DC}} + k_{DC}^{n_{DC}})}\right) - N(D)\right)/\tau_D \tag{4.8}$$

$$\frac{dE}{dt} = \left(\left(\frac{N(C)^{n_{EC}}}{(N(C)^{n_{EC}} + k_{EC}^{n_{EC}})} \times \frac{N(D)^{n_{ED}}}{(N(D)^{n_{ED}} + k_{ED}^{n_{ED}})}\right) - N(E)\right)/\tau_E \tag{4.9}$$

Para este problema, quando evoluído por regressão simbólica via Programação Genética através da biblioteca DEAP, também considerando 20 execuções independentes, obteve-se um erro mínimo de 122,98. As EDOs completas e os resultados da regressão simbólica estão disponíveis no material suplementar no repositório.

4.5.2 Ritmo Circadiano com 10 espécies

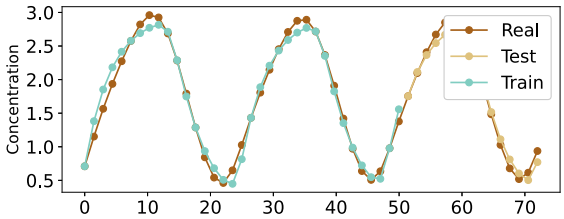
O segundo modelo gerado é para o ritmo circadiano de 10 espécies, onde cada gene é nomeado como A, ..., J, representando MP, P0, P1, P2, MT, T0, T1, T2, C, e CN, respectivamente, do modelo esquemático apresentado na Figura 46 na Seção 4.1.

Os dados na forma de séries temporais, discretizados e diagrama de transição de estados são apresentados na Figura 56a, 56b e 56c, respectivamente. Onze estados diferentes e 12 transições de estado foram determinados. Neste experimento houve ambiguidades nas transições de estados (1111111111 and 0000000000) e a forma de tratamento é apresentada na Figura 57.

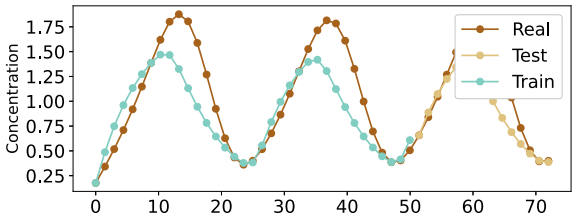
O mesmo procedimento aplicado ao problema com 5 espécies é aplicado aqui. As expressões lógicas obtidas para modelo são apresentadas na Tabela 26. Novamente, essas expressões lógicas foram utilizadas para determinar os BooleCubes e posteriormente convertidos em HillCubes, cujos comportamentos temporais estão disponíveis no material suplementar no repositório. Os gráficos dos dados reais e preditos pelo modelo são apresentados nas Figuras 58 e 59. Dentre as 20 execuções independentes, esse modelo alcançou um erro mínimo de 50.87. Para este problema, quando evoluído por regressão simbólica via Programação Genética através da biblioteca DEAP, também considerando as 20 execuções independentes, obteve-se um erro mínimo de 156,05. As EDOs completas e os resultados da regressão simbólica estão disponíveis no material suplementar no repositório.

Figura 55 – Resultados para as variáveis A, B e C para o ritmo circadiano de 5 variáveis. Os rótulos *Real, Train* e *Test* representam os dados reais, preditos no treino e preditos no teste, respectivamente.

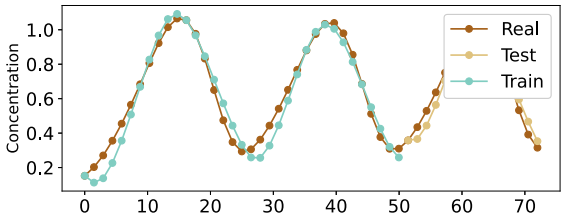
(a) Variável A: $\tau = 1,53, n = 15$ e $k = 0,74$.



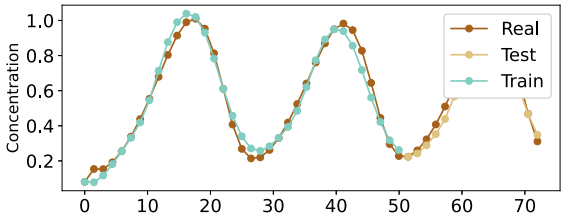
(b) Variável B: $\tau = 3,86, n = 6$ e $k = 0,59$.



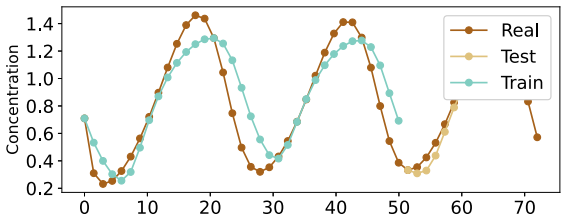
(c) Variável C: $\tau = 4,63, nCB = 15, nCC = 3, kCB = 0,53$ e $kCC = 1$.



(d) Variável D: $\tau = 3,64, nDB = 3, nDC = 5, kDB = 0,64$ e $kDC = 0,91$.



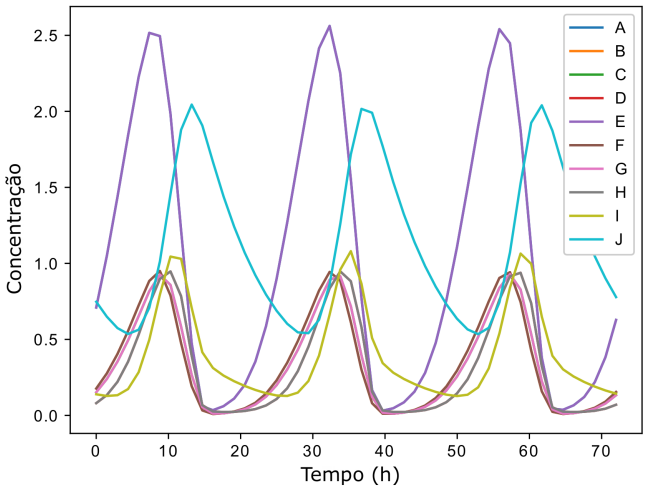
(e) Variável E: $\tau = 1,92, nEC = 10, nED = 3, kEC = 0,1$ e $kED = 0,36$.



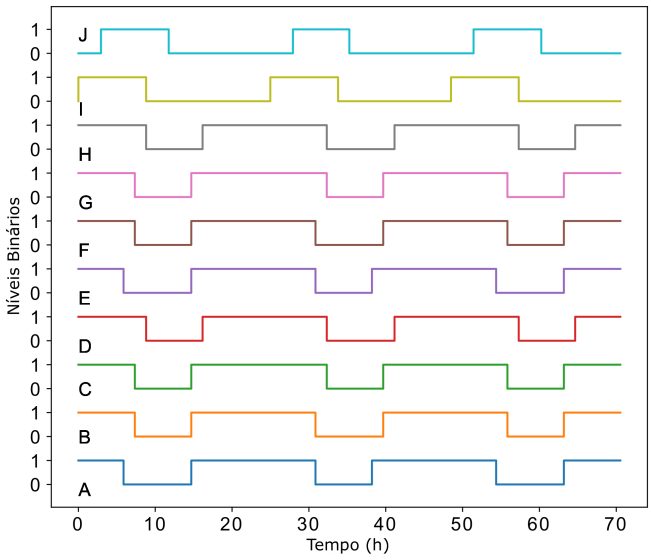
Fonte: Elaborado pelo autor (2023).

Figura 56 – Gráficos e diagrama de transição de estados para o problema do ritmo circadiano de 10 variáveis.

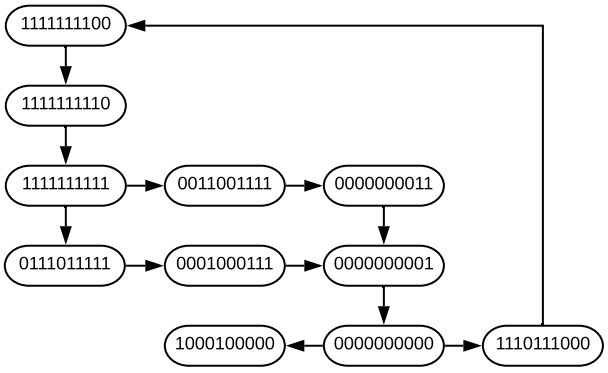
(a) Dados originais.



(b) Dados discretizados.



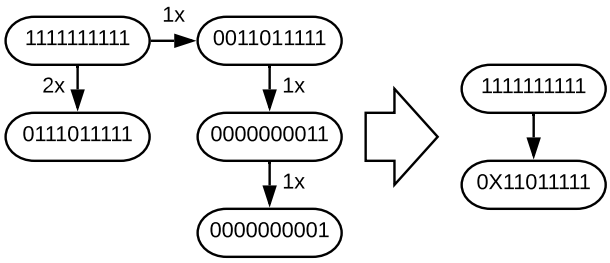
(c) Diagrama de transição de estados inicial.



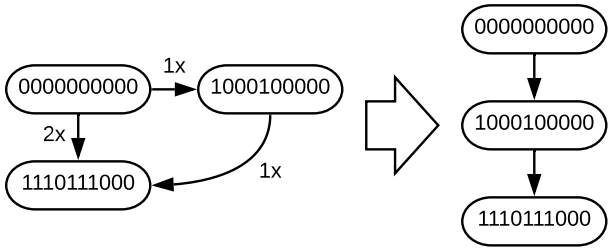
Fonte: SILVA *et al.* (2020).

Figura 57 – Ambiguidades e diagrama de transição de estados para o ritmo circadiano de 10 variáveis.

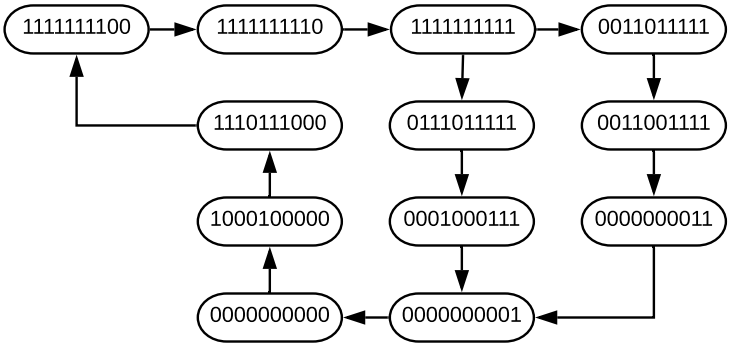
(a) Transições ambíguas de 1111111111, o número de ocorrências de cada transição e a transição utilizada.



(b) Transições ambíguas of 0000000000, o número de ocorrências de cada transição e a transição utilizada.



(c) Diagrama de transição de estados final.



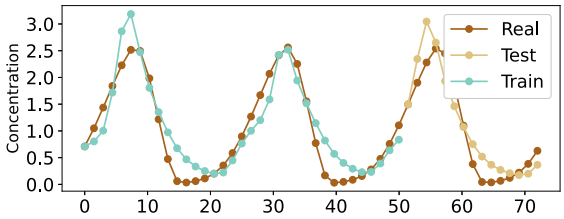
Fonte: SILVA *et al.* (2020).

4.6 AVALIAÇÃO DAS ETAPAS DO MÉTODO PROPOSTO

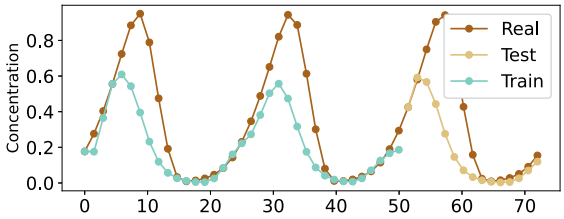
Como consequência do desenvolvimento do *framework* CGPGRN, dos estudos da literatura e dos resultados obtidos a partir dos problemas *benchmark* descritos anteriormente, uma série de análises sobre as etapas do método proposto foram realizadas a fim de não só aprimorar o método, mas também encontrar e caracterizar problemas associados à inferência de GRNs utilizando dados oriundos de perfilamento scRNA-Seq. As seções seguintes apresentam discussões, resultados e análises sobre as etapas de (i) pré-processamento, (ii) discretização, (iii) agrupamento, e (iv) inferência do modelo Booleano. É importante ressaltar que as seções apresentadas aqui não correspondem, necessariamente, à ordem cronológica de desenvolvimento das etapas, e nem baseado na qualidade dos resultados,

Figura 58 – Resultados para o ritmo circadiano de 10 variáveis. Os rótulos *Real, Train* e *Test* representam os dados reais, preditos no treino e preditos no teste, respectivamente.

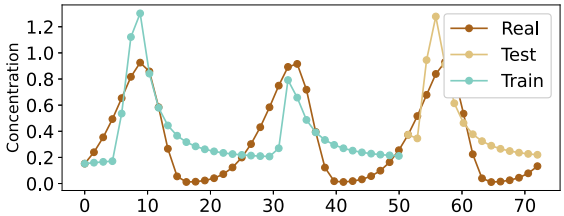
(a) Variável A: $\tau = 2{,}06$, $n_{AB} = 21$, $n_{AH} = 13$, $k_{AB} = 0{,}90$ e $k_{AH} = 0{,}16$.



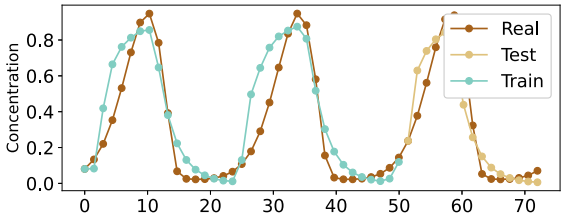
(b) Variável B: $\tau = 4{,}63$, $n_{BB} = 16$ $n_{BF} = 23$, $k_{BB} = 0{,}1$ e $k_{BF} = 0{,}34$.



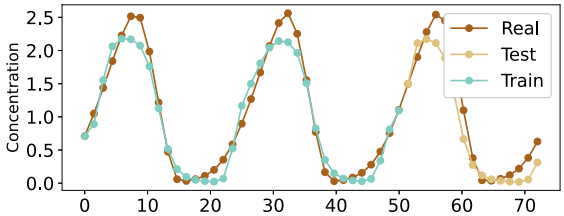
(c) Variável C: $\tau = 3,95$, $n_{CA} = 15$, $n_{CC} = 11$, $k_{CA} = 0{,}87$ e $k_{CC} = 1$.



(d) Variável D: $\tau = 4{,}78$, $n_{DB} = 3$, $n_{DD} = 12$, $n_{DE} = 4$, $n_{DH} = 14$, $k_{DB} = 0{,}1$, $k_{DD} = 0{,}36$, $k_{DE} = 0{,}21$ e $k_{DH} = 0{,}55$.



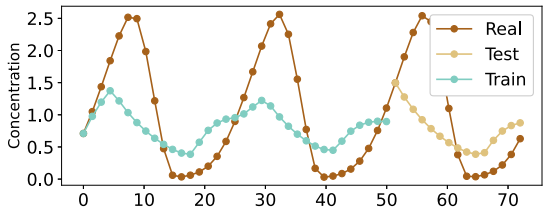
(e) Variável E: $\tau = 1{,}23$, $n_{EH} = 13$ e $k_{EH} = 0{,}39$.



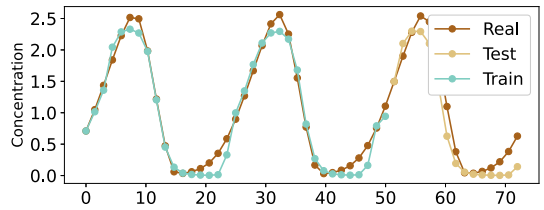
Fonte: Elaborado pelo autor (2023).

Figura 59 – Resultados para o ritmo circadiano de 10 variáveis. Os rótulos *Real, Train* e *Test* representam os dados reais, preditos no treino e preditos no teste, respectivamente.

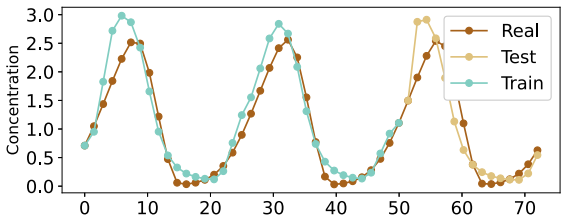
(a) Variável F: $\tau = 4,44$, $n_{FA} = 22$, $n_{FC} = 23$, $n_{FI} = 7$, $k_{FA} = 1$, $k_{FC} = 0{,}77$ e $k_{FI} = 0{,}36$.



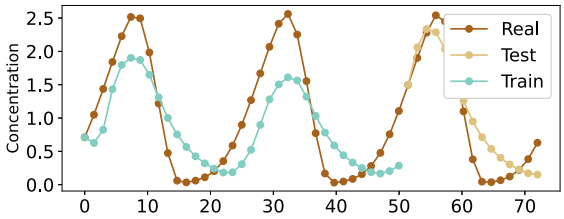
(b) Variável G: $\tau = 1,09$, $n_{GD} = 20$, $n_{GE} = 22$, $n_{GH} = 23$, $k_{GD} = 0{,}71$, $k_{GE} = 0{,}22$, e $k_{GH} = 0{,}46$.



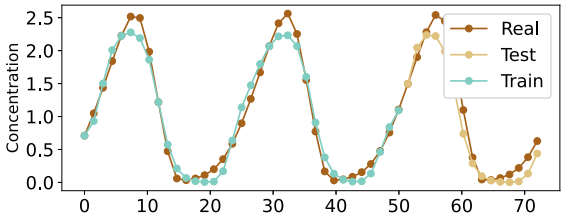
(c) Variável H: $\tau = 1{,}24$, $n_{HB} = 13$, $n_{HH} = 9$, $k_{HB} = 1$ e $k_{HH} = 0{,}23$.



(d) Variável I: $\tau = 3,47$, $n_{IB} = 21$, $n_{ID} = 3$, $n_{IE} = 24$, $n_{IH} = 19$, $k_{IB} = 0{,}31$, $k_{ID} = 0{,}66$, $k_{ID} = 0{,}90$ e $k_{IH} = 0{,}17$



(e) Variável J: $\tau = 2{,}06$, $n_{JB} = 23$, $n_{JF} = 15$, $k_{JB} = 0{,}22$ e $k_{JF} = 1$.



Fonte: Elaborado pelo autor (2023).

Tabela 26 – Expressões lógicas para o ritmo circadiano de 10 variáveis.

Variável	Expressão
A	B + H
B	(B xor F)
C	A + C
D	B$\overline{E}$H + B$\overline{D}$H + BDE$\overline{H}$
E	H
F	A$\overline{C}$I + A$\overline{I}$ + C$\overline{I}$
G	$\overline{E}$H + $\overline{D}$H + DEH
H	H + B
I	B$\overline{E}$H + B$\overline{D}$H + BDE$\overline{H}$
J	(B xor F)

Fonte: Elaborado pelo autor (2023).

mas sim, à sequência lógica do fluxograma da proposta, apresentada na Figura 27.

4.6.1 Pré-Processamento

Uma parte da etapa de pré-processamento é o uso de *smoothing splines* para a suavização dos dados de expressão gênica a fim de tratar as variações técnicas e biológicas bem como os *dropouts* oriundos da tecnologia de perfilamento por scRNA-Seq. Experimentos computacionais foram realizados a fim de determinar a importância do uso dessa suavização dos dados e sua relação com a qualidade das GRNs inferidas. Contudo, como o método de discretização também afeta o resultado final, o uso do *smoothing spline* é avaliado sob diversas técnicas de discretização. Os métodos de discretização considerados aqui são Bikmeans, EFD, EWD, Gallo, Kmeans (tanto no escopo de dados linha quanto matriz), Max -X%Max, Mean, Median, Top%X e TSD, descritas na Seção **2.11.1**. Todas as discretizações foram realizadas utilizando a ferramenta GEDPROTOOLS e as implementações da CGP estão disponíveis publicamente[9].

Os problemas considerados para essa análise são da categoria acurados, por possuírem mais de um *pseudotime*, para todas as taxas de *dropout*, apresentados na Seção 4.1, exceto o problema GSD. Os resultados foram obtidos utilizando o *framework* BEELINE (PRATAPA *et al.*, 2020), considerando AUPRC e AUROC. Para todos os experimentos, considera-se $n_c = l_b = 100$ e $n_r = 1$ e o número máximo de avaliações da função objetivo de 50.000.

Como cada *pseudotime* dá informação sobre uma possível trajetória celular, para cada *pseudotime* uma GRN é inferida. Além disso, como ressaltado anteriormente, a GRN final é obtida através da união das GRNs parciais obtidas para cada *pseudotime*. Nesta união, relações regulatórias iguais, obtidas em *pseudotimes* diferentes, são removidas e

[9] `https://github.com/ciml/cilamce2021_cgp-grn-discretization`

somente as relações regulatórias mais fortes são mantidas. Essa GRN final tem as relações regulatórias ranqueadas, da mais forte para a mais fraca, e então, avaliada.

As Tabelas 27, 28, 29 apresentam os resultados da mediana de AUPRC e AUROC dos métodos de discretização considerando o uso e o não uso do *smoothing spline* para os problemas mCAD, HSC e VSC, respectivamente. Boxplots adicionais sobre as análises do uso do pré-processamento para todos os problemas, em todas as taxas de *dropout*, tanto para AUPRC quanto para AUROC estão disponíveis no material suplementar no repositório.

Tabela 27 – Resultados da comparação do uso de *smooting splines* para o problema mCAD. Os melhores resultados são apresentados em negrito. R.Diff(%) apresenta a diferença relativa com o referencial NoSpline.

Método	AUPRC			AUROC		
	Spline	NoSpline	R,Diff(%)	Spline	NoSpline	R,Diff(%)
Bikmeans	**0,6299**	0,6079	+3,62	**0,4835**	0,3709	+30,4
EFD	**0,6959**	0,6407	+8,62	**0,5604**	0,4835	+15,9
EWD	0,6194	**0,6284**	-1,43	0,4341	**0,4478**	-3,10
Gallo	**0,6055**	0,6044	+0,18	**0,4368**	0,4231	+3,24
KmeansLinha	**0,7016**	0,6419	+9,30	**0,5412**	0,4258	+27,1
KmeansMatrix	**0,7597**	0,5695	+33,4	**0,6923**	0,3791	+82,6
Max50	**0,7071**	0,6285	+12,5	**0,5467**	0,4588	+19,2
Max54	0,6319	**0,6471**	-2,35	**0,4451**	0,4066	+9,47
Max60	0,5997	**0,6028**	-0,51	0,4148	**0,4396**	-5,64
Mean	**0,6338**	0,6225	+1,82	**0,4725**	0,4423	+6,83
Median	**0,6857**	0,6407	+7,02	**0,5604**	0,4835	+15,9
Top25	0,5401	**0,6810**	-20,7	0,3571	**0,4945**	-27,8
Top30	0,5480	**0,6938**	-21,0	0,3599	**0,5577**	-35,5
Top35	0,5347	**0,6252**	-14,5	0,3132	**0,4780**	-34,5
Top40	0,5672	**0,6360**	-10,8	0,3736	**0,4698**	-20,5
Top45	0,6002	**0,6251**	-3,92	0,4560	**0,4698**	-2,94
Top50	0,6209	**0,6407**	-3,10	**0,4918**	0,4808	+2,29
Top55	**0,5864**	0,5709	+2,72	**0,4121**	0,3764	+9,48
Top60	**0,6368**	0,6235	+2,13	0,4451	**0,4505**	-1,20
TSD	**0,6414**	0,6269	+2,31	0,4505	**0,4670**	-3,53

Fonte: Adaptado de SILVA *et al.* (2021).

De acordo com a Tabela 27 é possível perceber que, para o problema mCAD, os valores de AUPRC e AUROC variam significativamente a depender do método de discretização empregado. Em contagem absoluta, o uso do *spline* fornece os melhores resultados em 11/20 situações quando considerada ambas AUPRC e AUROC. Ao analisar as diferenças relativas, percebe-se também grande variabilidade, sendo de -21,0% a 33,4% para AUPRC e -35,5% a 82,62% para AUROC. Não usar o *spline* é melhor, principalmente, quando considerados os métodos de discretização *Top* com parâmetros [25%, 50%] e *Max*

com parâmetros 54 e 60, tanto para AUPRC quanto AUROC. Contudo, utilizar o *spline* melhorou significativamente a maioria dos casos, chegando a 33,4% e 82,6% para AUPRC e AUROC, respectivamente, quando considerada a discretização KmeansMatrix. Além disso, é importante ressaltar que nos experimentos de avaliação do *framework* CGPGRN, apresentadas na Seção 4.2, o método de discretização utilizado foi o Bikmeans. Para esse método, os resultados são melhores quando utiliza-se o *spline*, tanto para AUPRC quanto para AUROC.

Tabela 28 – Resultados da comparação do uso de *smooting splines* para o problema HSC. Os melhores resultados são apresentados em negrito. R.Diff(%) apresenta a diferença relativa com o referencial NoSpline.

Método	AUPRC			AUROC		
	Spline	NoSpline	R.Diff(%)	Spline	NoSpline	R.Diff(%)
Bikmeans	**0,3058**	0,2700	+13,3	**0,5915**	0,5844	+1,21
EFD	**0,2556**	0,2364	+8,12	**0,5492**	0,5000	+9,84
EWD	**0,2786**	0,2507	+11,1	**0,5492**	0,5227	+5,07
Gallo	0,2473	**0,2504**	-1,23	0,5048	**0,5120**	-1,41
KmeansLinha	**0,2677**	0,2619	+2,21	0,5338	**0,5453**	-2,11
KmeansMatrix	**0,2961**	0,2733	+8,34	**0,5723**	0,5717	+0,10
Max50	**0,278**	0,2651	+4,87	**0,5388**	0,5283	+1,99
Max54	**0,268**	0,2518	+6,43	0,5287	**0,5332**	-0,84
Max60	**0,2471**	0,2453	+0,73	0,5042	**0,5197**	-2,98
Mean	**0,2823**	0,2397	+17,8	**0,5579**	0,4971	12,23
Median	**0,2525**	0,2364	+6,81	**0,5326**	0,5000	6,46
Top25	**0,2377**	0,2364	+0,55	**0,5074**	0,5000	1,48
Top30	**0,2457**	0,2364	+3,93	**0,5206**	0,5000	4,12
Top35	**0,2555**	0,2364	+8,08	**0,5301**	0,5000	6,02
Top40	**0,2603**	0,2364	+10,1	**0,5354**	0,5000	7,08
Top45	**0,2574**	0,2364	+8,88	**0,5429**	0,5000	8,58
Top50	**0,2452**	0,2364	+3,72	**0,5237**	0,5000	4,74
Top55	**0,2557**	0,2364	+8,16	**0,5388**	0,5000	7,76
Top60	0,2529	**0,2590**	-2,35	**0,5341**	0,5259	1,56
TSD	0,2312	**0,2364**	-2,20	0,4884	**0,5000**	-2,32

Fonte: Adaptado de SILVA *et al.* (2021).

Para o problema HSC, cujos resultados estão apresentados na Tabela 28, apesar de também haver variabilidade nos valores de AUROC e AUPRC, na maioria dos casos o uso do *spline* melhora a qualidade das soluções. Em contagem absoluta, o uso do *spline* fornece os melhores resultados em 17/20 situações para AUPRC, com diferença relativa variando entre -2,35% e 17,8%, e em 15/20 situações para AUROC, com diferença relativa entre -2,32% e 12,23%. A maior melhoria foi observada quando considerada a discretização *Mean*. Para este problema, em relação à discretização Top%X, os resultados são melhores em todos os casos quando considerado o *spline*, exceto para o parâmetro

60. É interessante perceber que os valores sem suavização para essa discretização são exatamente os mesmos para os parâmetros [25%, 55%] tanto para AUPRC quanto para AUROC e que a suavização forneceu melhorias entre 0,55% e 10,1% para AUPRC e entre 1,48% e 8,58% para AUROC. Novamente, é importante ressaltar que os resultados são melhores quando utiliza-se *spline* para o método Bikmeans.

Tabela 29 – Resultados da comparação do uso de *smooting splines* para o problema VSC. Os melhores resultados são apresentados em negrito. Resultados não obtidos são representados com "-". R.Diff(%) apresenta a diferença relativa com o referencial NoSpline.

Método	AUPRC			AUROC		
	Spline	NoSpline	R.Diff(%)	Spline	NoSpline	R.Diff(%)
Bikmeans	0,2506	**0,2521**	-0,60	0,4785	**0,4813**	-0,58
EFD	**0,3211**	0,2679	+19,9	**0,5813**	0,4951	+17,4
EWD	**0,2992**	0,2679	+11,7	**0,5557**	0,4951	+12,2
Gallo	**0,2886**	0,2522	+14,4	**0,5398**	0,4724	+14,3
KmeansLinha	**0,2783**	-	-	**0,5154**	-	-
KmeansMatrix	0,2205	**0,2418**	-8,81	0,4337	**0,4508**	-3,79
Max50	**0,2762**	0,2573	+7,35	**0,5004**	0,4780	+4,69
Max54	**0,2732**	0,2630	+3,88	0,4874	**0,4915**	-0,83
Max60	**0,2799**	0,2767	+1,16	**0,5118**	0,5106	+0,24
Mean	**0,3087**	0,2872	+7,49	**0,5646**	0,4911	+15,0
Median	**0,3211**	0,2413	+33,1	**0,5829**	0,4679	+24,6
Top25	0,1932	**0,2670**	-27,6	0,3346	**0,4667**	-28,3
Top30	0,2019	**0,2527**	-20,1	0,3467	**0,4630**	-25,1
Top35	0,2272	**0,2531**	-10,2	0,4236	**0,4614**	-8,19
Top40	**0,2501**	0,2393	+4,51	**0,4793**	0,4370	+9,68
Top45	**0,3052**	0,2609	+17,0	**0,5720**	0,4866	+17,6
Top50	**0,3127**	0,2573	+21,5	**0,5724**	0,4772	+20,0
Top55	**0,2810**	0,2592	+8,41	**0,5301**	0,4805	+10,3
Top60	**0,2761**	0,2299	+20,1	**0,5037**	0,3850	+30,8
TSD	**0,3014**	-	-	**0,5512**	-	-

Fonte: Adaptado de SILVA *et al.* (2021).

Por fim, a Tabela 29 apresenta os resultados para o problema VSC. Usar o *spline* foi melhor em 15/20 situações para AUPRC e 14/20 para AUROC. Além disso, é importante ressaltar que dois métodos de discretização (KmeansLinha e TSD) não encontraram relações regulatórias corretas sem o uso do *spline*. As melhorias quando considerado o uso da suavização chegam a 33% para AUPRC e quase 31% para AUROC. O fato observado para o *Top%* no problema mCAD também está presente no VSC. Contudo, somente para os parâmetros [25%, 35%]. Além disso, o Bikmeans apresentou melhores resultados quando não considerada a etapa de suavização. Contudo, essa diferença é menor que 1%.

Em números absolutos, para todos os problemas, usar o *spline* forneceu melhores resultados em 43/60 situações para AUPRC e em 40/60 situações para AUROC.

De maneira geral, para o método de discretização *Top%X* os resultados são melhores quando não utiliza-se o *spline*. Isso também foi observado em alguns casos para os métodos *Max* e *EWD*. É importante ressaltar que os modelos acurados são derivados de modelos Booleanos da literatura. Esses modelos geralmente utilizam um *threshold* para determinar os pontos de expressão gênica que serão discretizados em nível lógico alto e baixo. Em relação ao método *Top*, sua aplicação considera os maiores valores de expressão gênica. Por esse motivo, é possível que o *threshold* do modelo Booleano original esteja em aproximadamente 50% para o problema mCAD, e 35% para o problema VSC. Isso justifica o melhor resultado obtido sem o uso da suavização, uma vez que os valores muito altos de expressão gênica podem ser eliminados neste processo. Com isso, o ponto de corte (*threshold*) será um valor menor do que o do dado original, sem o *spline*. Essa mesma explicação é válida para os métodos de discretização *Max* e *EWD*, conforme definição destes métodos apresentada na Seção **2.11.1**.

Esses resultados reforçam a importância em utilizar o pré-processamento proposto na Seção 3.1.

4.6.2 Agrupamento

Conforme apresentado na Seção 3.2, a etapa de agrupamento é não obrigatória e tem como objetivo auxiliar na seleção de subconjuntos de genes que compartilhem relações regulatórias. Contudo, diversas técnicas de agrupamento podem ser utilizadas. Experimentos computacionais foram realizados para avaliar o desempenho do *framework* CGPGRN utilizando diferentes abordagens de agrupamento, a saber: Kendall Tau, Spearman, Pearson, *Clustering* Aglomerativo e KMeans. Além disso, para a abordagem KMeans foram testadas três possibilidades: a evolução de cada gene separadamente, a evolução de um circuito completo por *cluster* (denominada *Full TT*), e uma discretização por *cluster*, denominada FD. A discretização por *cluster* é analisada pois os genes de um dado *cluster* podem ser discretizados considerando somente aqueles do mesmo *cluster* ou a discretização pode acontecer sobre o conjunto inteiro de dados. Para técnicas de discretização que levam em consideração apenas estatística (EFD, EWD, Gallo, Max, Mean, Median, Top e TSD), isso não faz diferença. Contudo, para o método Bikmeans utilizado aqui, o agrupamento aplicado em diferentes conjuntos de dados resultará em estados discretos diferentes. Para fins de comparação, considerou-se também não utilizar agrupamento (denotado aqui como *No Clustering (NC)*). A Tabela 30 resume o significado dos prefixos e sufixos utilizados.

Para os experimentos, foram considerados os problemas experimentais da configuração 500nTF e todas as redes de referência. Os resultados são comparados com os algoritmos estado da arte com melhor desempenho nos dados experimentais: GENIE3, GRNBOOST2, PIDC, SINCERITIES e PPCOR. Os hiperparâmetros (número de *clusters* ou *threshold*) foram determinados empiricamente através de experimentos preliminares,

Tabela 30 – Resumo da nomenclatura utilizada nos experimentos de agrupamento.

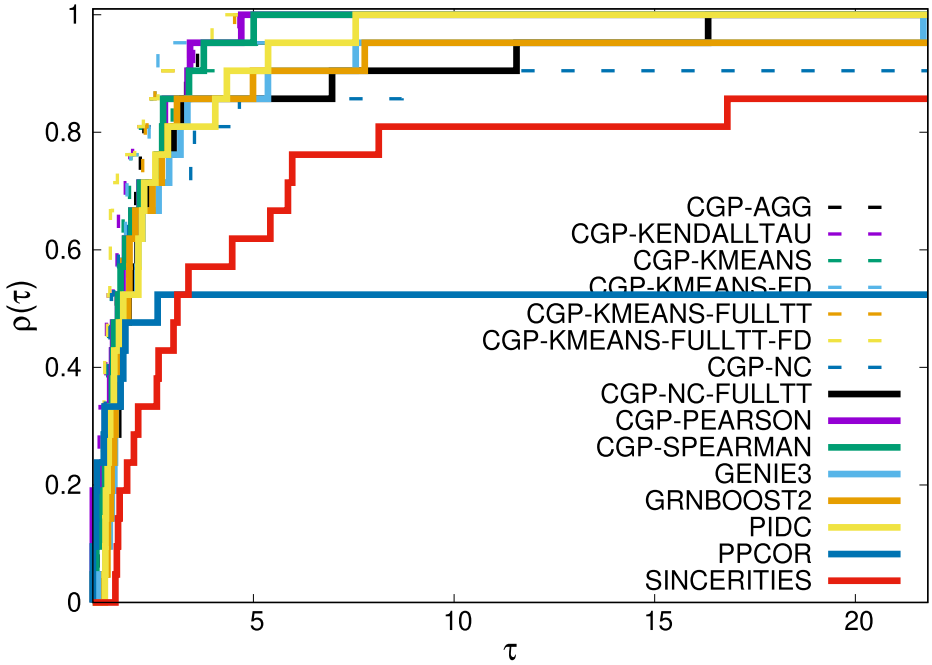
Termo	Tipo	Significado
FULLTT	Sufixo	Evolução de um circuito completo considerando todos os genes.
FD	Sufixo	A discretização é realizada considerando os genes de cada *cluster*.
AGG	Prefixo	Indica o uso do *clustering* hierárquico aglomerativo.
NC	Prefixo	Indica o não uso de *clustering*.

Fonte: Elaborado pelo autor (2024).

que indicaram um número máximo de *clusters* igual a 10 e um *threshold* de 0,7.

A Figura 60 apresenta os resultados dos PPs considerando todos os algoritmos e os métodos de discretização para o melhor caso. O caso da mediana está disponível no material suplementar no repositório. As áreas sob a curva dos PPs e a quantidade de vezes em que cada algoritmo obteve os melhores resultados são apresentadas na Tabela 31.

Figura 60 – PPs para o melhor caso considerando todos os métodos de agrupamento e os algoritmos estado da arte.



Fonte: Elaborado pelo autor (2024).

A partir dos resultados apresentados no PP, é possível concluir que: (i) CGP-PEARSON obteve os melhores resultados na maioria dos problemas (maior $\rho(1)$), (ii) CGP-KMEANS-FULLTT-FD é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) CGP-KMEANS-FULLTT-FD apresenta o melhor desempenho geral (maior área sob a curva do PP).

Contudo, ainda que CGP-PEARSON tenha obtido os melhores resultados na maioria dos problemas, de acordo com a Tabela 31, esse algoritmo obteve melhores

Tabela 31 – Áreas sob os PPs para o melhor caso e número de vezes em que cada algoritmo obteve os melhores resultados.

Método	Área	Contagem
CGP-KMEANS-FULLTT-FD	1,0	3
CGP-KENDALLTAU	0,9929	3
CGP-KMEANS-FD	0,9922	2
CGP-KMEANS-FULLTT	0,9904	2
CGP-PEARSON	0,9857	4
CGP-SPEARMAN	0,9823	0
CGP-AGG	0,9707	0
PIDC	0,9613	0
CGP-KMEANS	0,9457	3
GRNBOOST2	0,9211	0
CGP-NC-FULLTT	0,9193	0
GENIE3	0,9172	0
CGP-NC	0,8827	2
SINCERITIES	0,7531	0
PPCOR	0,5303	2

Fonte: Elaborado pelo autor (2024).

resultados em 4/21 casos. Já o CGP-KMEANS-FULLTT-FD obteve melhores resultados em 3/21 casos. Esse mesmo resultado é obtido para os algoritmos CGP-KMEANS e CGP-KENDALLTAU. Logo, como a diferença é de apenas uma unidade entre CGP-PEARSON e CGP-KMEANS-FULLTT-FD e que este segundo algoritmo apresentou o melhor desempenho geral e é a abordagem mais confiável, pode-se considerar o CGP-KMEANS-FULLTT-FD como a melhor alternativa. Além disso, é importante destacar que não utilizar o agrupamento é sempre pior do que qualquer abordagem que utiliza agrupamento, em relação ao desempenho geral. Ainda, utilizar FULLTT fornece melhores resultados quando não considera-se uma etapa de agrupamento. A abordagem que utiliza somente o KMeans, sem FULLTT e sem FD é a técnica de agrupamento que obteve os piores resultados. Quando considerados os algoritmos estado da arte, 7 abordagens da CGP com agrupamento superam o melhor algoritmo (PIDC). Esta abordagem só obteve resultados melhores que a CGP-KMEANS e as abordagens que não utilizaram agrupamento. SINCERITIES e PPCOR apresentaram resultados muito inferiores às técnicas de CGP.

Desta forma, é possível perceber que o uso de técnicas de agrupamento tendem a facilitar a obtenção de relações regulatórias corretas pelo algoritmo de busca. Contudo, conforme será discutido na Seção 4.7, as técnicas de agrupamento tendem a apresentar bons resultados quando o conjunto de dados utilizados não apresenta grandes quantidades de genes que compartilham relações regulatórias.

4.6.3 Discretização

Experimentos computacionais foram realizados a fim de determinar o impacto do método de discretização na qualidade das GRNs inferidas. Esta seção é dividida como segue: (i) discute-se a aplicação dos métodos tradicionais de discretização da literatura, apresentados na Seção **2.11.1**, (ii) apresentam-se resultados e discussões sobre o método de discretização proposto (DSSPD) e apresentados na Seção **3.3.1**, e (iii) mostra-se a possibilidade de combinar diferentes abordagens de discretização em métodos *ensemble*.

4.6.3.1 Análise dos métodos de discretização da literatura

Experimentos foram conduzidos para analisar o desempenho de diferentes abordagens de discretização, discutidos na Seção 3.3, quando aplicados à CGP como algoritmo de inferência, utilizando problemas *benchmark* perfilados por scRNA-Seq na forma de séries temporais. Os métodos de discretização considerados aqui são Bikmeans, EFD, EWD, Gallo, Kmeans (tanto no escopo de dados linha quanto matriz), Max -X%Max, Mean, Median, Top%X e TSD, descritas na Seção **2.11.1**. Todas as discretizações foram realizadas utilizando a ferramenta GEDPROTOOLS e as implementações da CGP estão disponíveis publicamente[10].

Os problemas considerados para essa análise são da categoria acurados, para todas as taxas de *dropout*, apresentados na Seção 4.1, exceto o problema GSD. Os resultados foram obtidos utilizando o *framework* BEELINE (PRATAPA *et al.*, 2020), considerando AUPRC e AUROC. Para todos os experimentos, considera-se $n_c = l_b = 100$ e $n_r = 1$ e o número máximo de avaliações da função objetivo de 50.000.

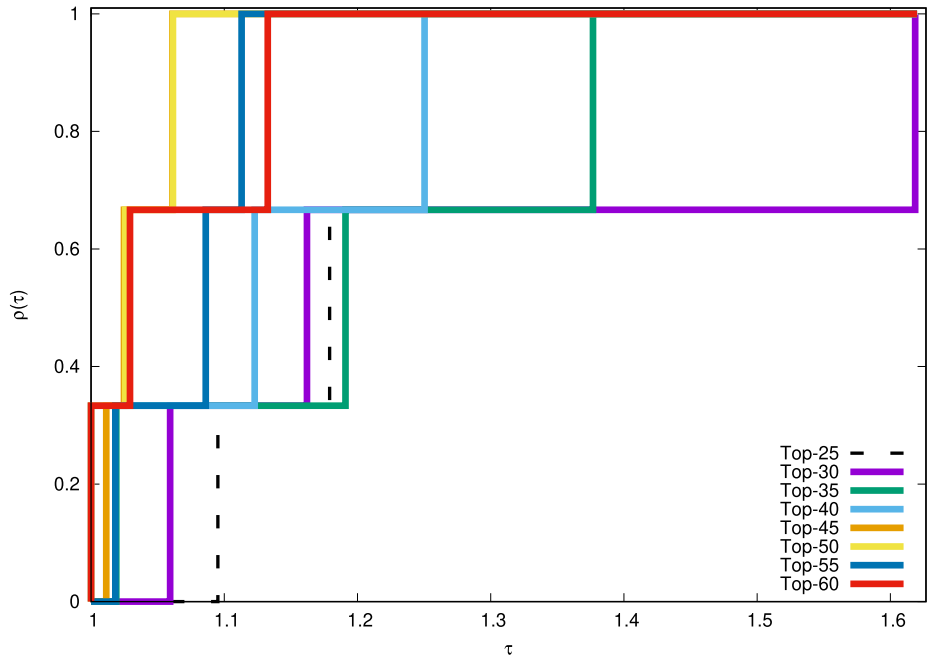
Como cada *pseudotime* dá informação sobre uma possível trajetória celular, uma GRN é inferida para cada *pseudotime*. Além disso, como ressaltado anteriormente, a GRN final é obtida através da união das GRNs parciais obtidas para cada *pseudotime*. Nesta união, relações regulatórias iguais, obtidas em *pseudotimes* diferentes são removidas e somente as relações regulatórias mais fortes são mantidas. Essa GRN final tem as relações regulatórias ranqueadas, da mais forte para a mais fraca, e então, avaliada.

Como os métodos Top%X e Max -X%Max possuem parâmetros, experimentos preliminares foram conduzidos para a determinação dos melhores valores. Para esta análise, consideramos somente os resultados com o uso de *smoothing splines* pois mostraram-se superiores, conforme discutido na Seção **4.6.1**. Para o Top%X, valores no intervalo [25, 60] com um passo de 5% foram considerados. Para o Max -X%Max, o valor de referência é 54% (MADEIRA; OLIVEIRA, 2005). Portanto, analisa-se os valores {50, 54, 60}. Os resultados para os parâmetros do Top%, para mediana de AUPRC e AUROC, são apresentados nos PPs das Figuras 61 e 62, respectivamente e os resultados para os

[10] `https://github.com/ciml/cilamce2021_cgp-grn-discretization`

parâmetros do Max -X%Max, para mediana de AUPRC e AUROC, são apresentados nos PPs das Figuras 63 e 64, respectivamente.

Figura 61 – *PPs* da mediana de AUPRC para os parâmetros do Top%. Áreas: Top-25 (0,5444), Top-30 (0,5743), Top-35 (0,7176), Top-40 (0,8383), Top-45 (0,9943), Top-50 (1,0), Top-55 (0,9264) e Top60 (0,9577).



Fonte: SILVA *et al.* (2021).

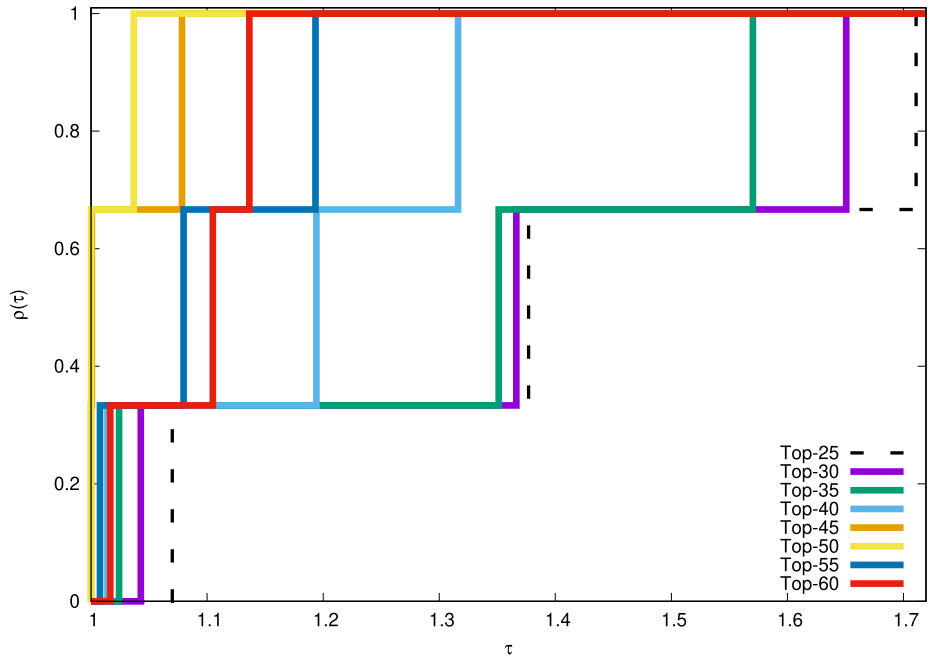
Para AUPRC do Top% é possível concluir que: (i) Top40, Top-50 e Top-60 encontraram os melhores resultados para a maioria dos problemas (maior $\rho(1)$), (ii) Top-50 é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) Top-50 obteve o melhor desempenho geral (maior área sob a curva).

Já para AUROC do Top%X é possível concluir que: (i) Top-50 encontrou os melhores resultados para a maioria dos problemas (maior $\rho(1)$), (ii) Top-50 é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) Top-50 obteve o melhor desempenho geral (maior área sob a curva).

Para AUPRC do Max -X%Max é possível concluir que: (i) Max-50 encontrou os melhores resultados para a maioria dos problemas (maior $\rho(1)$), (ii) Max-50 é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) Max-50 obteve o melhor desempenho geral (maior área sob a curva).

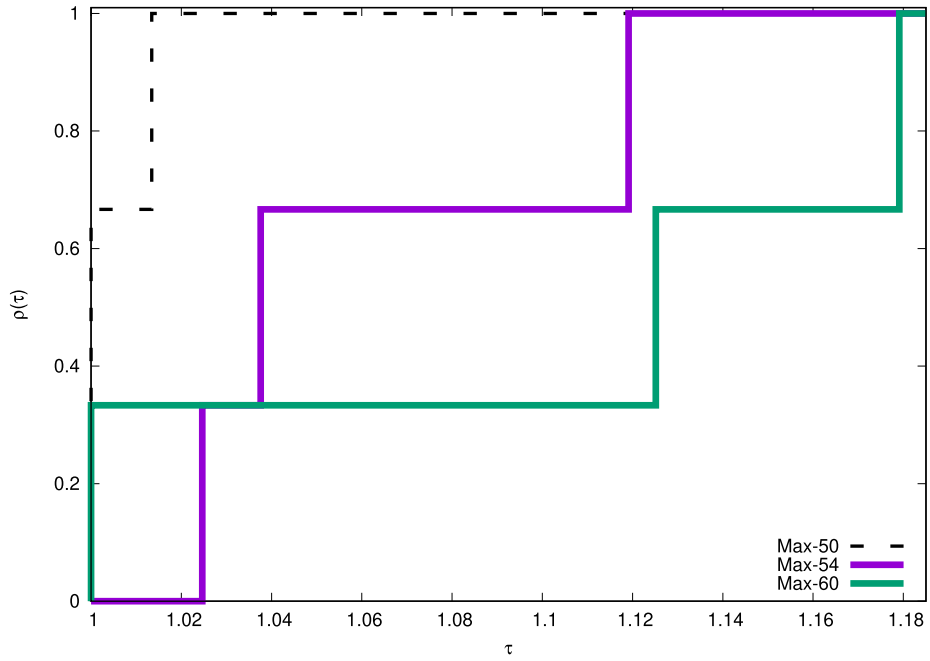
Já para AUROC do Max -X%Max é possível concluir que: (i) Max-50 encontrou os melhores resultados para a maioria dos problemas (maior $\rho(1)$), (ii) Max-50 é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) Max-50 obteve o melhor desempenho geral (maior área sob a curva). Resultados adicionais sobre os parâmetros são apresentados no material suplementar disponibilizado no repositório. Baseado nesses resultados, é possível

Figura 62 – *PPs* da mediana de AUROC para os parâmetros do Top%. Áreas: Top-25 (0,4651), Top-30 (0,5117), Top-35 (0,5663), Top-40 (0,7672), Top-45 (0,9798), Top-50 (1,0), Top-55 (0,8836) e Top60 (0,8945).



Fonte: SILVA *et al.* (2021).

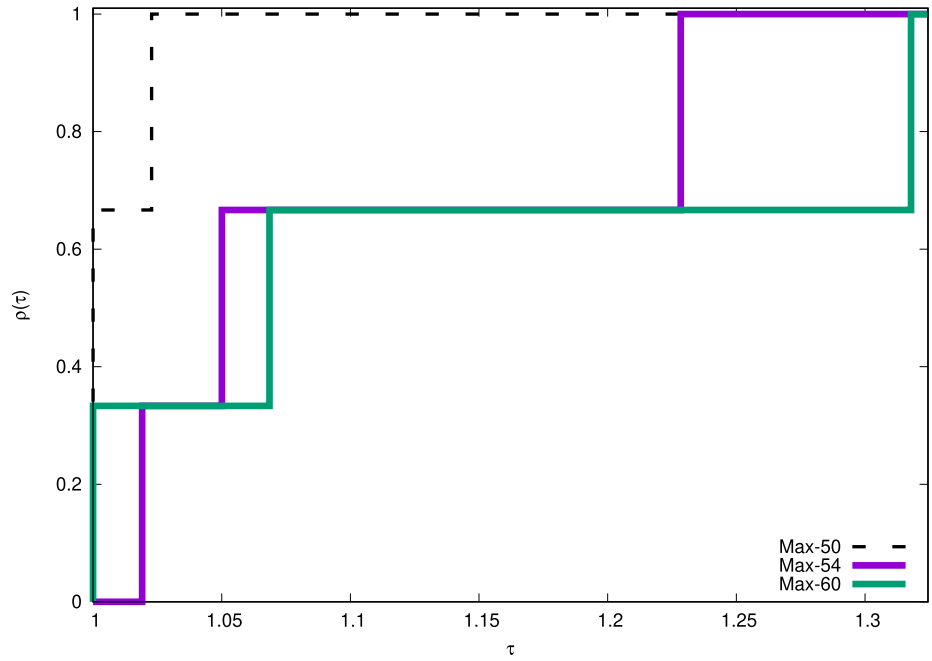
Figura 63 – *PPs* da mediana de AUPRC para os parâmetros do Max -X%Max. Áreas: Max-50 (1,0), Max-54 (0,6795) e Max-60 (0,4449).



Fonte: SILVA *et al.* (2021).

concluir que o melhor parâmetro para a discretização Top% é de 50% e para a discretização Max -X%Max também é de 50%.

Figura 64 – *PPs* da mediana de AUROC para os parâmetros do Max -X%Max. Áreas: Max-50 (1,0), Max-54 (0,7049) e Max-60 (0,6093).



Fonte: SILVA *et al.* (2021).

Agora, considerando todos os métodos de discretização e os melhores parâmetros determinados, as Figuras 65, 66, e 67 apresentam os resultados para AUPRC e AUROC para os problemas HSC, mCAD e VSC. Os *boxplots* apresentados nas Figuras 65a e 65b mostram que Bikmeans obteve melhor desempenho para o HSC.

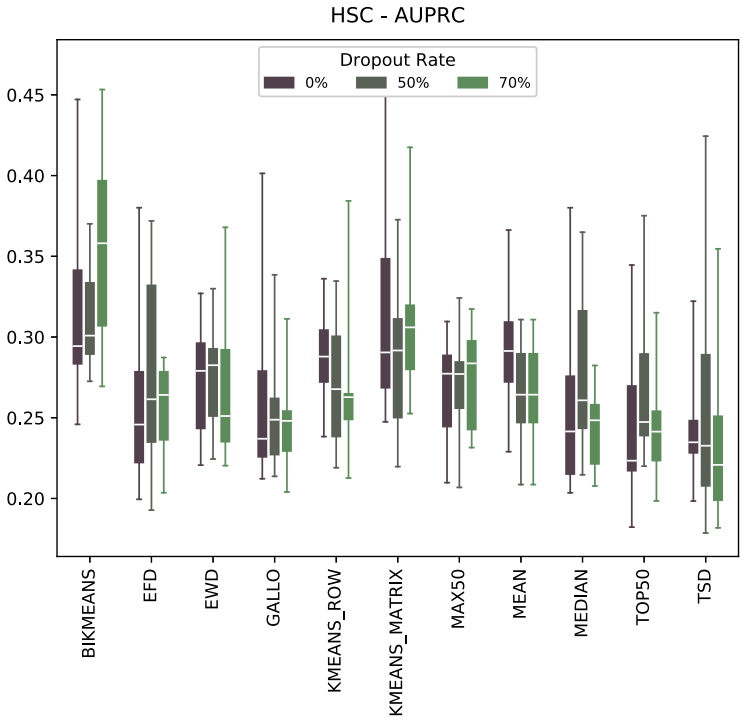
Ainda, mesmo com 70% de dropout, a CGP com Bikmeans obteve bons resultados. Contudo, para mCAD e VSC, o método EFD obteve melhores resultados na maioria dos casos. Além disso, todas as abordagens de discretização apresentaram grande variância, independentemente da taxa de *dropout*. Para todos os problemas, quando a abordagem Mean é considerada, os resultados obtidos com 50% e 70% de *dropout* são o mesmo. Em geral, *dropouts* não afetaram substancialmente o desempenho da CGP, diferentemente do que é relatado na literatura (PRATAPA *et al.*, 2020). Uma possível explicação é que os *dropouts* são considerados como caso de irrelevância nas tabelas verdade, facilitando a CGP em obter soluções factíveis.

As Figuras 68 e 69 apresentam as PPs para AUPRC e AUROC, respectivamente, considerando todas as estratégias de discretização e os melhores parâmetros definidos previamente.

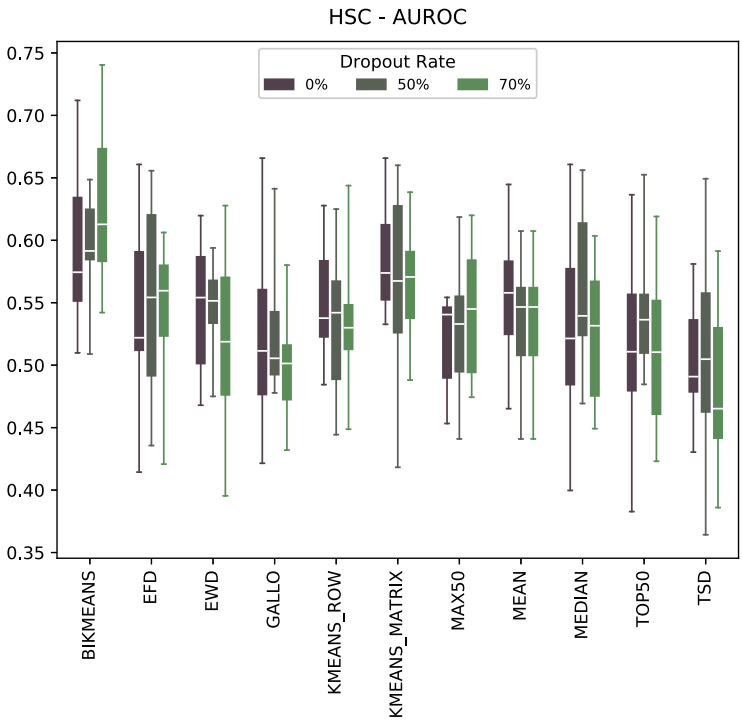
Para AUPRC, é possível concluir que (i) Median obteve os melhores resultados para a maioria dos problemas (maior $\rho(1)$), (ii) KmeansRow é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), e (iii) EFD obteve o melhor desempenho geral (maior área sob a curva).

Figura 65 – Resultados para o problema HSC considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do HSC



(b) Resultados para AUROC do HSC

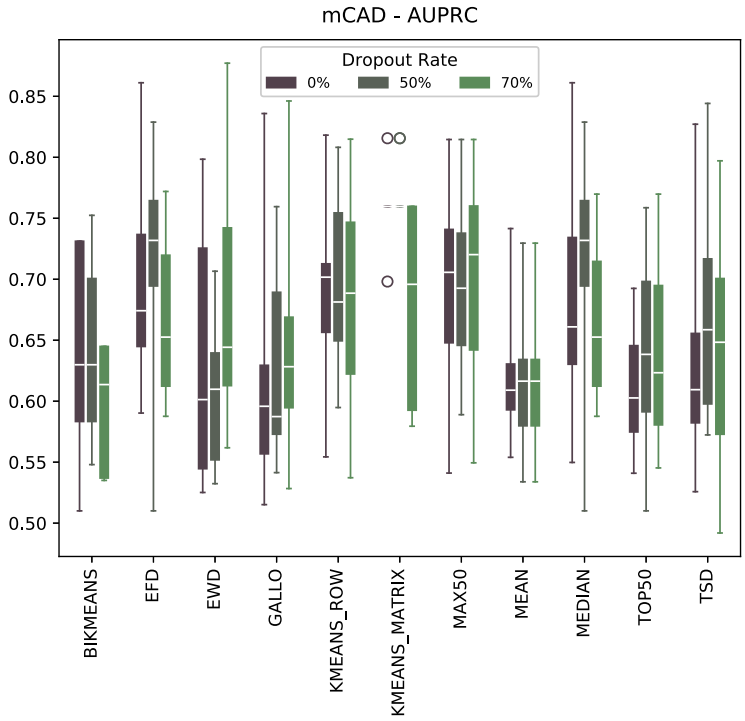


Fonte: SILVA *et al.* (2021).

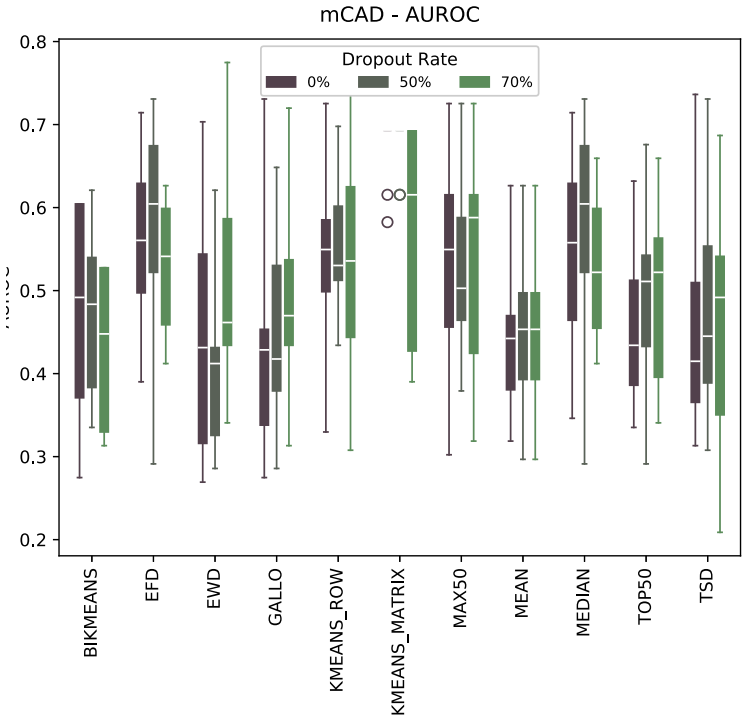
Quando considerada a AUROC, conclui-se: (i) Median e EFD são abordagens que encontraram os melhores resultados para a maioria dos problemas, (ii) Median é a

Figura 66 – Resultados para o problema mCAD considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do mCAD



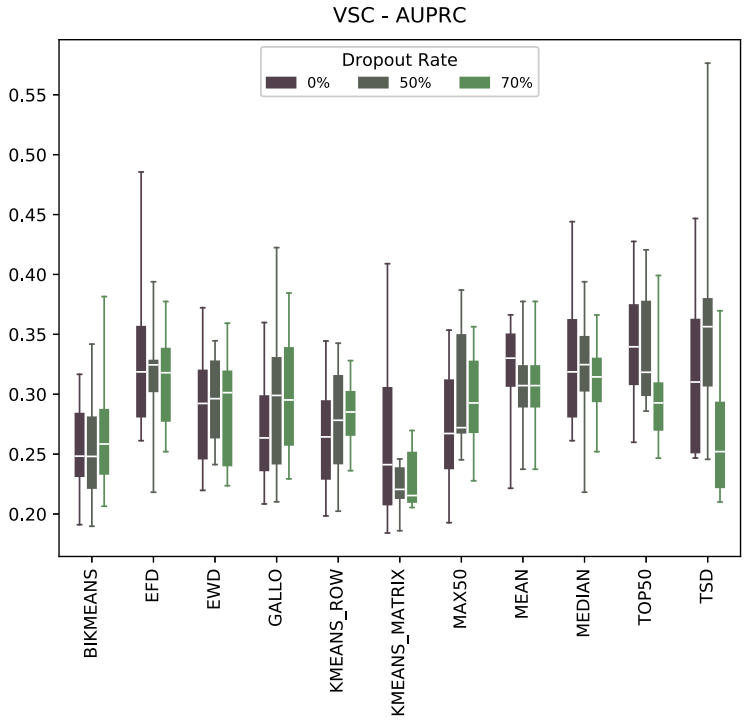
(b) Resultados para AUROC do mCAD



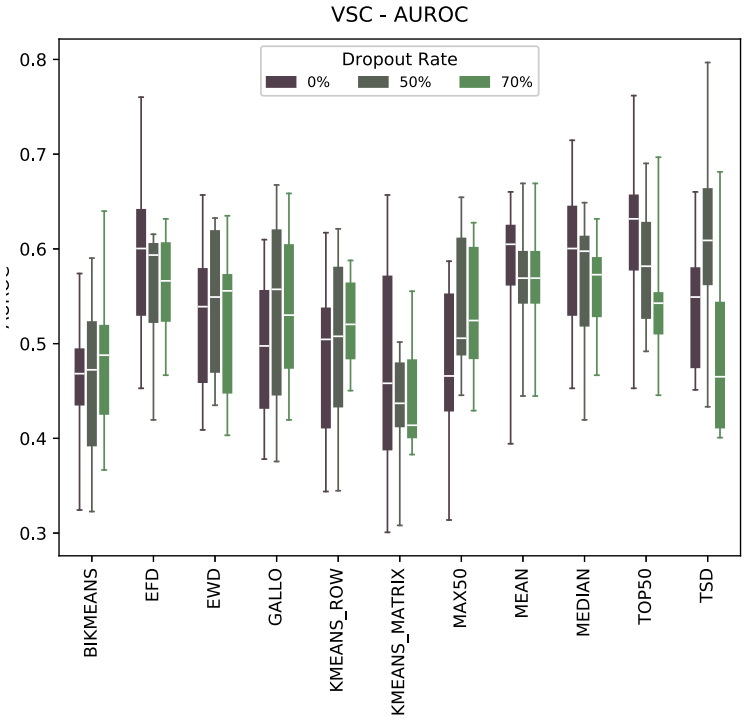
Fonte: SILVA *et al.* (2021).

Figura 67 – Resultados para o problema VSC considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do VSC

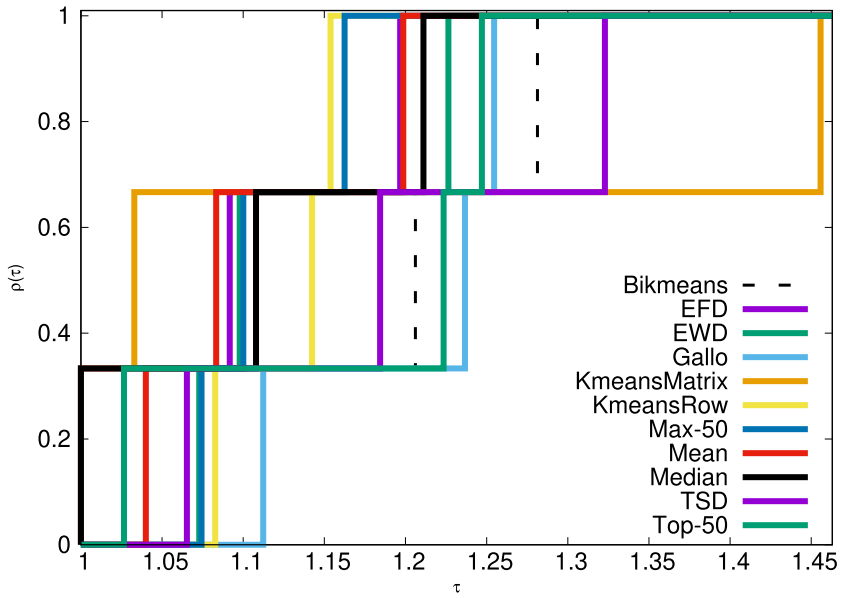


(b) Resultados para AUROC do VSC



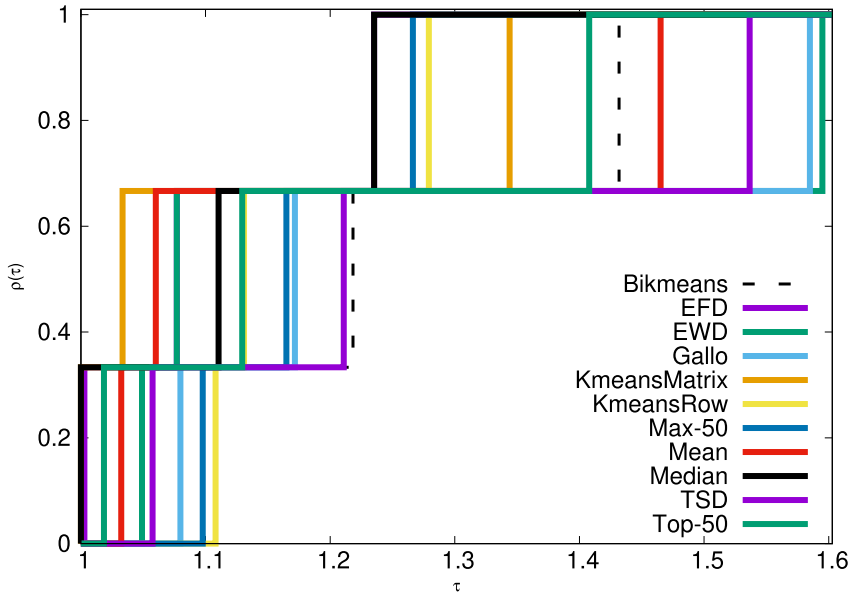
Fonte: SILVA *et al.* (2021).

Figura 68 – PPs para AUPRC considerando todos os métodos de discretização.



Fonte: SILVA *et al.* (2021).

Figura 69 – PPs para AUROC considerando todos os métodos de discretização.



Fonte: SILVA *et al.* (2021).

abordagem mais confiável, e (iii) EFD obteve o melhor desempenho geral.

Logo, pode-se concluir que EFD obteve o melhor desempenho geral em ambas AUPRC e AUROC. Contudo, Median é uma boa escolha e obteve o segundo melhor desempenho geral.

4.6.3.2 Análise do DSSPD

Experimentos computacionais foram realizados a fim de avaliar o desempenho do método de discretização proposto na Seção **3.3.1** e seu impacto sobre a qualidade das GRNs inferidas. Como o DSSPD possui um parâmetro μ_{var}, os experimentos são conduzidos em três etapas: (i) análise de sensibilidade de parâmetros do DSSPD, (ii) comparação entre o DSSPD e a CGP com Bikmeans, e (iii) análise comparativa da CGP-DSSPD com os algoritmos estado da arte PPCOR, SINCERITIES, GRNBOOST2, GENIE3 e PIDC. A justificativa para o uso do Bikmeans no experimento (ii) é que tal técnica de discretização é a mais utilizada na literatura e apresentava bons resultados (LI *et al.*, 2010; GALLO *et al.*, 2015), além de ter sido utilizada como referência nos nossos experimentos antes do desenvolvimento do DSSPD.

Para todos os experimentos, o conjunto de funções da CGP é $\Gamma = \{$AND, OR, NOT, XOR, XNOR, NAND, NOR$\}$, $\mu = 1$, $\lambda = 4$, lb $= n_c$, e $n_r = 1$, como sugerido em (MILLER, 2011) e conforme experimentos anteriores. O número de avaliações da função objetivo e o número de nós (n_c) é variado, tendo em vista a utilização de diferentes conjuntos de dados. Dessa forma, para os dados acurados foi considerado $n_c = 500$ e 50.000 como número máximo de avaliações da função objetivo. Já para os problemas experimentais, adotou-se $n_c = 1.000$ e 2.000.000 de avaliações da função objetivo.

Como ressaltado anteriormente, para os problemas que possuem mais de um *pseudotime*, uma GRN é inferida para cada trajetória e a rede final é obtida a partir da união das GRNs parciais. Além disso, por conta da natureza estocástica da CGP, cinco execuções independentes são realizadas para cada experimento.

Para os problemas da categoria acurados foram avaliados AUPRC e AUROC e foi utilizado a EP para os problemas experimentais, conforme realizado em (PRATAPA *et al.*, 2020). Todos os códigos fonte estão disponíveis publicamente[11] e resultados adicionais estão disponibilizados no material suplementar no repositório.

Primeiramente, estudos preliminares indicaram um valor de $\mu_{var} = 0,02$ para o DSSPD. A partir desse valor de referência, uma análise de sensibilidade de parâmetros é realizada para $\mu_{var} \in \{0,005, 0,01, 0,02, 0,05\}$. As Tabelas 32, 33 e 34 apresentam os resultados para os problemas acurados considerando todas as taxas de *dropout* de 0%, 50% e 70%, respectivamente. Valores negativos indicam que o parâmetro de comparação gerou resultados piores do que a referência (0,02).

Para a configuração de 0% de *dropout*, conforme apresentado na Tabela 32, é possível perceber que a referência obtém melhores resultados para 6/12 casos quando considerada a AUPRC e para 7/12 casos para AUROC. Para os problemas GSD, HSC e VSC, a referência apresenta melhores resultados na maioria dos casos. Quando a referência

[11] `https://github.com/jeduaardo/biosystems_dsspd`

Tabela 32 – Análise de sensibilidade de parâmetros para o melhor caso considerando 0% de dropout. Os valores apresentados são a diferença relativa entre o parâmetro μ_{var} da segunda coluna e a referência de 0,02.

		0% *dropout*			
	μ_{var}	GSD	HSC	mCAD	VSC
AUPRC	0,005	3,2%	0,2%	0,5%	-8,2%
	0,01	-0,9%	-4,9%	0,5%	-12,8%
	0,05	-2,4%	-1,1%	0,9%	2,6%
AUROC	0,005	0,4%	-6,3%	6,0%	-6,6%
	0,01	-1,7%	-1,3%	6,0%	-7,6%
	0,05	-2,4%	-2,5%	6,0%	0,6%

Fonte: Adaptado de SILVA *et al.* (2024).

não é melhor, a diferença percentual é pequena. Contudo, para o problema mCAD, ainda que a diferença seja inferior a 1%, a referência é pior em todos os casos de AUPRC. Já para AUROC, essa diferença permaneceu constante em 6%.

Tabela 33 – Análise de sensibilidade de parâmetros para o melhor caso considerando 50% de dropout. Os valores apresentados são a diferença relativa entre o parâmetro μ_{var} da segunda coluna e a referência de 0.02.

		50% *dropout*			
	μ_{var}	GSD	HSC	mCAD	VSC
AUPRC	0,005	-0,3%	18,9%	0,3%	-4,0%
	0,01	-1,2%	8,0%	5,1%	-1,5%
	0,05	4,3%	9,0%	8,2%	2,9%
AUROC	0,005	1,6%	8,2%	11,3%	-3,5%
	0,01	0,09%	7,7%	14,0%	-5,1%
	0,05	3,1%	2,6%	17,2%	0,06%

Fonte: Adaptado de SILVA *et al.* (2024).

Já para a Tabela 33, os resultados para 50% de *dropout* indicam que a referência é o melhor valor para os problemas GSD e VSC. Para o problema mCAD, principalmente quando considerada a AUROC, todos os demais parâmetros superam o valor de referência em mais de 10%. Já para o problema HSC, tanto para AUPRC quando para AUROC, o valor de referência é pior em todos os casos, variando entre 8% e 18.9% para AUPRC e entre 2.6% e 8.2% para AUROC.

Por fim, de acordo com a Tabela 34, quando considerada a taxa de 70% de *dropout*, o valor de referência apresenta o melhor resultado em relação a todos os parâmetros, tanto para AUPRC quando AUROC para o problema VSC. Para o problema GSD, a diferença de AUPRC em relação à referência é pequena e para AUROC é quase nula. Já para os

Tabela 34 – Análise de sensibilidade de parâmetros para o melhor caso considerando 70% de dropout. Os valores apresentados são a diferença relativa entre o parâmetro μ_{var} da segunda coluna e a referência de 0.02.

	μ_{var}	70% *dropout* GSD	HSC	mCAD	VSC
AUPRC	0,005	3,1%	10,0%	4,3%	-16,1%
	0,01	1,0%	2,2%	2,9%	-17,5%
	0,05	0,0%	-4,9%	-4,8%	-2,0%
AUROC	0,005	0,02%	3,8%	8,6%	-6,9%
	0,01	0,02%	-0,02%	6,4%	-5,1%
	0,05	0,00%	-1,2%	-7,7%	-1,5%

Fonte: Adaptado de SILVA *et al.* (2024).

problemas HSC e mCAD, o valor de referência só é melhor em relação ao parâmetro de 0.05.

De maneira geral, o valor de referência apresentou resultados piores para o problema mCAD, especialmente para a taxa de 50% de dropout. Por sua vez, a referência é melhor quando considerado 70% de *dropout* em relação à $\mu_{var} = 0.05$. Os melhores resultados são obtidos pelo valor de referência principalmente para 0% de dropout (7/12 casos) e para 70% (6/12 casos). É importante ressaltar que para todas as taxas de *dropout* a referência apresenta os melhores resultados tanto para AUPRC quanto para AUROC no problema VSC. Já para o problema GSD, os parâmetros não apresentaram diferença significativa, com resultados variando entre 0.9% a 2.4% em melhorias e 0% e 3.2% em piora. Além disso, o problema VSC é o que apresenta maior número de *pseudotimes* (5). Já o problema GSD é o problema com maior quantidade de genes (19). Tendo em vista que esses são problemas acurados e que os dados experimentais tipicamente apresentam altas taxas de *dropout* e que é comum utilizar perfilamento por scRNA-Seq para a análise de diferenciação celular, os resultados indicam que o parâmetro $\mu_{var} = 0.02$ é uma boa escolha.

Contudo, é possível que a grande variação nos resultados seja ocasionada pelo número de estados identificados durante a discretização. Menores valores de μ_{var} tendem a gerar mais estados, uma vez que a suavidade da discretização é menor. Isso significa dizer que pequenas variações do nível de expressão gênica serão consideradas para serem discretizadas nos pontos de corte ao invés de manter o nível lógico atribuído para o intervalo anterior. De maneira similar, quanto maior o parâmetro, menor o número de estados. Uma baixa quantidade de estados também pode ser um problema, já que a grande quantidade de situações de irrelevância aumentam o espaço de busca do algoritmo de inferência.

Para a segunda parte da experimentação, utilizando o valor de referência $\mu_{var} = 0.02$ derivado do experimento anterior, os resultados entre a CGP tradicional utilizando Bikmeans (mesma configuração dos experimentos apresentados na Seção 4.2) são comparados

com a CGP com o DSSPD, nomeados de CGP e CGP-DSSPD, respectivamente.

As Tabelas 35 e 36 apresentam os resultados para AUPRC e AUROC considerando os melhores resultados para os problemas acurados, contendo o pior caso (menor avaliação), o primeiro quartil (Q1), a média, mediana, terceiro quartil (Q3), melhor caso (maior avaliação) e desvio padrão (Std).

Tabela 35 – Valores de AUPRC para o melhor caso. Melhores resultados estão apresentados em negrito.

	Prob.	Alg.	Pior	Q1	Média	Mediana	Q3	Melhor	Std.
0% dropout	GSD	CGP	0,2141	0,2257	0,2402	0,2408	0,2539	0,2669	**0,0173**
		CGP-DSSPD	**0,2371**	**0,2647**	**0,2726**	**0,2718**	**0,2894**	**0,3047**	0,0207
	HSC	CGP	**0,2195**	**0,2469**	**0,2812**	**0,2776**	**0,3063**	**0,3755**	**0,0431**
		CGP-DSSPD	0,2115	0,2444	0,2651	0,2689	0,2854	0,3084	**0,0280**
	mCAD	CGP	0,5684	0,6113	0,6200	0,6219	0,6406	0,6451	**0,0237**
		CGP-DSSPD	**0,7400**	**0,7666**	**0,8113**	**0,8177**	**0,8562**	**0,8834**	0,0492
	VSC	CGP	0,2544	0,2709	0,3075	0,2885	0,3504	0,3767	**0,0450**
		CGP-DSSPD	**0,2642**	**0,3398**	**0,3725**	**0,3516**	**0,4083**	**0,4902**	**0,0644**
50% dropout	GSD	CGP	0,2090	0,2267	0,2391	0,2386	0,2569	0,2726	**0,0205**
		CGP-DSSPD	**0,2266**	**0,2562**	**0,2759**	**0,2680**	**0,2947**	**0,3260**	0,0294
	HSC	CGP	0,2138	**0,2718**	**0,2899**	**0,2982**	**0,3192**	**0,3451**	0,0403
		CGP-DSSPD	**0,2201**	0,2435	0,2554	0,2464	0,2743	0,3130	**0,0277**
	mCAD	CGP	0,5701	0,6424	0,6443	0,6498	0,6609	0,6818	**0,0292**
		CGP-DSSPD	**0,6747**	**0,7327**	**0,7863**	**0,7652**	**0,8704**	**0,8834**	0,0745
	VSC	CGP	0,2638	0,2943	0,3308	0,3142	0,3271	0,4912	**0,0631**
		CGP-DSSPD	**0,2744**	**0,3261**	**0,3764**	**0,3507**	**0,4094**	**0,5146**	0,0708
70% dropout	GSD	CGP	**0,2207**	0,2273	0,2388	0,2361	0,2442	0,2664	**0,0142**
		CGP-DSSPD	0,1932	**0,2550**	**0,2606**	**0,2597**	**0,2737**	**0,3055**	0,0274
	HSC	CGP	**0,2442**	0,2474	**0,2759**	0,2542	**0,2990**	**0,3394**	0,0360
		CGP-DSSPD	0,2341	**0,2622**	0,2754	**0,2751**	0,2946	0,3137	**0,0251**
	mCAD	CGP	0,5626	0,6619	0,6991	0,6816	0,7345	0,8539	0,0765
		CGP-DSSPD	**0,7242**	**0,7425**	**0,7794**	**0,7606**	**0,8086**	**0,8702**	**0,0459**
	VSC	CGP	**0,2789**	0,2995	0,3349	0,3302	0,3455	0,4515	**0,0489**
		CGP-DSSPD	0,2630	**0,3227**	**0,3862**	**0,3680**	**0,4683**	**0,5146**	0,0866

Fonte: SILVA *et al.* (2024).

De acordo com a Tabela 35 é possível perceber que a proposta é melhor em 10/16 casos quando considerados os valores de mediana para AUPRC. Além disso, para 0% de dropout, a proposta apresenta melhorias que variam de 3.13% a 31.48% na mediana. Para 50% de *dropout* as melhorias estão entre 11.62% e 17.76%. Exceto para o problema HSC onde a CGP com Bikmeans obteve melhores resultados. Já para a configuração de 70% de dropout, a proposta sempre supera a CGP tradicional, com melhorias entre 8.22% e 11.59%.

Já para a Tabela 36, a proposta também é melhor em 10/16 casos em relação à mediana para AUROC. Quando considerada taxa de *dropout* de 0% a proposta só não

Tabela 36 – Valores de AUROC para o melhor caso. Melhores resultados estão apresentados em negrito.

	Prob.	Alg.	Pior	Q1	Média	Mediana	Q3	Melhor	Std.
0% dropout	GSD	CGP	0,4981	0,5281	0,5380	0,5408	0,5500	0,5700	**0,0194**
		CGP-DSSPD	**0,4993**	**0,5561**	**0,5622**	**0,5698**	**0,5812**	**0,6115**	0,0338
	HSC	CGP	0,4636	**0,5389**	**0,5743**	**0,5793**	**0,5919**	**0,7090**	0,0615
		CGP-DSSPD	**0,4728**	0,5365	0,5572	0,5739	0,5822	0,5907	**0,0356**
	mCAD	CGP	0,3791	0,4286	0,4566	0,4588	0,4945	0,5000	**0,0383**
		CGP-DSSPD	**0,5604**	**0,6387**	**0,6747**	**0,6429**	**0,7294**	**0,8077**	0,0757
	VSC	CGP	0,4715	0,5140	0,5586	0,5488	0,5929	0,6659	0,0574
		CGP-DSSPD	**0,5382**	**0,5703**	**0,6049**	**0,6142**	**0,6354**	**0,6772**	**0,0423**
50% dropout	GSD	CGP	0,4846	0,5277	0,5357	0,5343	0,5612	0,5649	**0,0269**
		CGP-DSSPD	**0,5206**	**0,5530**	**0,5660**	**0,5608**	**0,5897**	**0,6082**	0,0272
	HSC	CGP	0,4670	**0,5718**	**0,5888**	**0,6061**	**0,6319**	**0,6470**	0,0571
		CGP-DSSPD	**0,4927**	0,5286	0,5524	0,5477	0,5677	0,6445	**0,0422**
	mCAD	CGP	0,3791	0,4904	0,4951	0,5000	0,5302	0,5549	0,0492
		CGP-DSSPD	**0,5000**	**0,5783**	**0,6588**	**0,6071**	**0,7857**	**0,8077**	**0,1104**
	VSC	CGP	0,4927	0,5474	**0,5917**	0,5801	0,6065	0,7325	0,0673
		CGP-DSSPD	**0,5041**	**0,5547**	0,5902	**0,6175**	**0,6238**	**0,6423**	**0,0455**
70% dropout	GSD	CGP	**0,5122**	0,5203	0,5372	0,5389	0,5493	0,5618	**0,0176**
		CGP-DSSPD	0,4655	**0,5383**	**0,5545**	**0,5615**	**0,5827**	**0,6061**	0,0384
	HSC	CGP	**0,5167**	0,5322	**0,5679**	0,5417	**0,6125**	**0,6575**	0,0503
		CGP-DSSPD	0,5156	**0,5420**	0,5671	**0,5758**	0,5862	0,6085	**0,0301**
	mCAD	CGP	0,3571	0,5206	0,5742	0,5714	0,6277	0,7802	0,1049
		CGP-DSSPD	**0,5604**	**0,6003**	**0,6462**	**0,6401**	**0,6745**	**0,7857**	**0,0618**
	VSC	CGP	0,4854	0,5506	0,5857	0,5951	**0,6335**	**0,6537**	0,0548
		CGP-DSSPD	**0,5041**	**0,5541**	**0,5911**	**0,6089**	0,6191	**0,6537**	**0,0500**

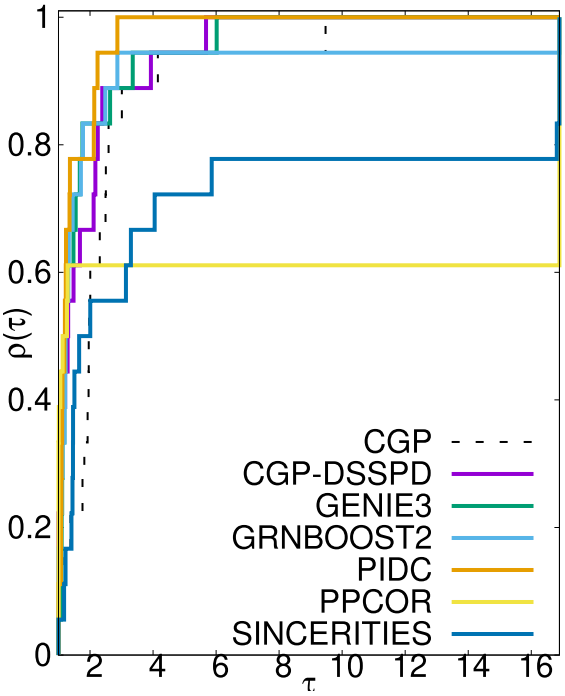
obteve melhores resultados para o problema HSC. Contudo, a diferença entre os resultados é inferior a 1%. No geral, para essa configuração, as melhorias são de 5,36% para GSD, 11,92% par VSC e 40,13% para mCAD. Para 50% de dropout, o comportamento é similar à 0% de dropout, onde as melhorias estão entre 4.96% e 21,4%, exceto para o problema HSC, piorando em torno de 9,64%. Por fim, para a taxa de 70% de *dropout*, a proposta obtém melhores resultados para todos os problemas, com melhorias entre 2,32% e 12,02%. As maiores melhorias foram encontradas para o problema mCAD, em todas as configurações de *dropout*.

Dessa forma, é possível concluir que a proposta é capaz de obter estados discretos que aumentam consideravelmente os resultados quando utilizando a CGP como algoritmo de busca para a inferência de GRNs. Além disso, as melhorias são obtidas especialmente quando considerada a taxa de 70% de dropout, o que reforça as vantagens da proposta tendo em vista a alta taxa de *dropout* presente em dados experimentais por conta da natureza da tecnologia de perfilamento por scRNA-Seq.

O último experimento consiste na comparação entre a proposta e os algoritmos

estado da arte. Também foi incluída a CGP com Bikmeans a fins de comparação. Para esta comparação, conforme mostrado nos experimentos anteriores, o parâmetro $\mu_{var} = 0,02$ foi adotado. O PP da Figura 70 apresenta os resultados obtidos em relação à mediana e o da Figura 71 em relação ao melhor caso.

Figura 70 – PP para a mediana nos problemas experimentais. Áreas: PIDC (1.0), GENIE3 (0.9799), CGP-DSSPD (0.9744), GRNBOOST2 (0.9446), CGP (0.9314), SINCERITIES (0.7387), e PPCOR (0.6250).



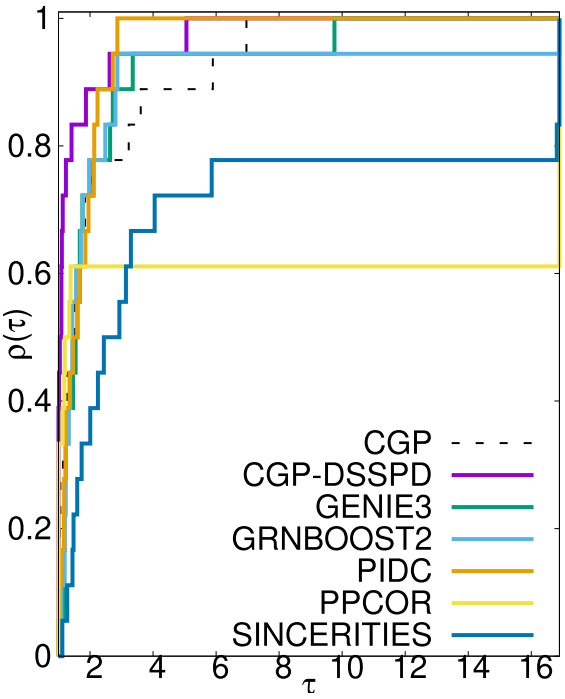
Fonte: SILVA *et al.* (2024).

Quando considerando a mediana, é possível concluir que: (i) CGP-DSSPD obteve os melhores resultados na maioria dos problemas (maior $\rho(1)$), (ii) PIDC é a abordagem mais confiável (menor τ tal que $\rho(\tau) = 1$), seguido pela CGP-DDSPD, e (iii) PIDC apresenta o melhor desempenho geral (maior área sob a curva do PP), seguido por GENIE3 e CGP-DSSPD, respectivamente. A diferença entre o GENIE3 e o CGP-DSSPD acontece apenas na terceira casa decimal.

Já para o caso dos melhores valores, percebe-se que: (i) CGP-DSSPD obteve os melhores resultados na maioria dos problemas, (ii) PIDC é a abordagem mais confiável, seguida pelo CGP-DSSPD, e (iii) CGP-DSSPD tem o melhor desempenho geral, seguido por PIDC e GENIE3.

Esses resultados ressaltam que a proposta é o algoritmo com melhor desempenho quando considerados os melhores valores e é competitivo com os algoritmos estado da arte tanto no caso da mediana quanto no melhor caso. Além disso, é possível perceber nos PPs que a proposta apresenta desempenho superior que a CGP com Bikmeans para ambos casos de mediana e melhor valor.

Figura 71 – PP para o melhor caso nos problemas experimentais. CGP-DSSPD (1.0), PIDC (0.9846), GENIE3 (0.9535), CGP (0.9533), GRNBOOST2 (0.9319), SINCERITIES (0.7249), e PPCOR (0.6236).



Fonte: SILVA *et al.* (2024).

4.6.3.3 Análise de método Ensemble

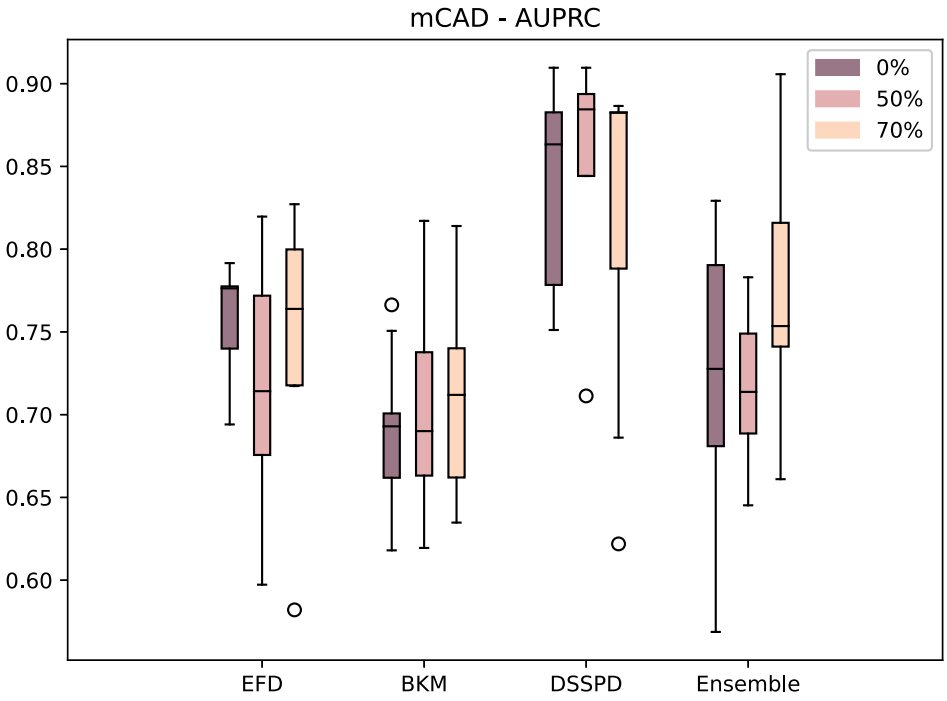
Tendo em vista o bom resultado apresentado pelo método de discretização EFD para problemas da categoria acurados, experimentos computacionais preliminares foram realizados a fim de avaliar a eficiência deste método combinado com a abordagem utilizando Bikmeans e a proposta (DSSPD) em um *ensemble*. A discretização é realizada em todo o *dataset* de expressão gênica e o estado lógico de cada ponto no tempo é determinado por votação. Desta forma, por maioria simples, o estado lógico com maior número de votos é atribuído para aquele ponto no tempo. Os três problemas acurados utilizados nos experimentos de avaliação de métodos de discretização da literatura, apresentados na Seção *4.6.3.1* também são considerados aqui. Para CGP, também foram adotados os mesmos parâmetros do experimento anterior, a saber $n_c = l_b = 100$ e $n_r = 1$ e o número máximo de avaliações da função objetivo de 50.000. O mesmo procedimento de união das redes parciais para obtenção da GRN final é adotado aqui.

Os resultados de AUPRC e AUROC para os problemas mCAD, VSC e HSC são apresentados nas Figuras 72, 73 e 74, respectivamente, onde 0%, 50% e 70% representam as taxas de *dropout*. Além disso, BKM é a sigla utilizada para Bikmeans.

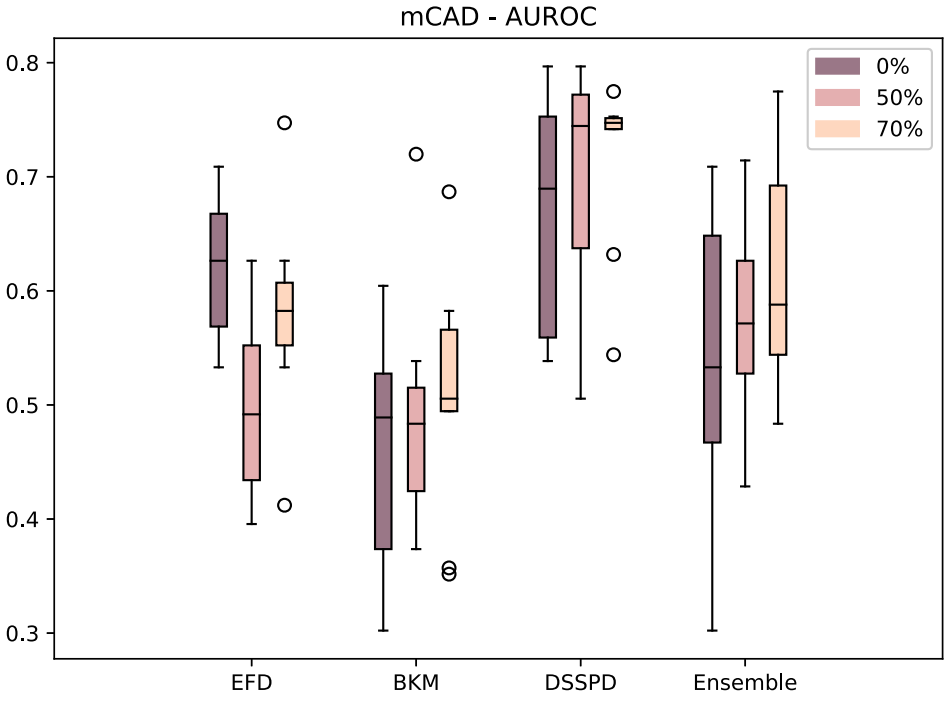
Para o problema mCAD, é possível perceber que, conforme esperado, o EFD apresenta melhores resultados que o Bikmeans, tanto para AUPRC quanto para AUROC. Contudo, o DSSPD apresentou resultados melhores que todas as abordagens, em todas as

Figura 72 – Resultados para o problema mCAD considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do mCAD



(b) Resultados para AUROC do mCAD



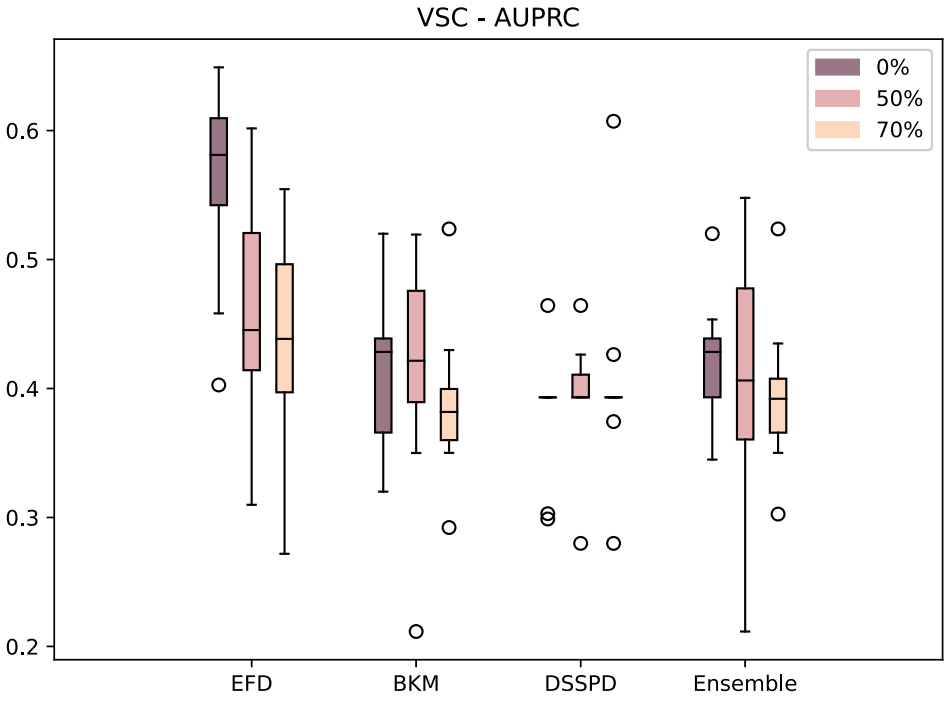
Fonte: Elaborado pelo autor (2024).

taxas de *dropout*. A combinação dos três métodos em um método ensemble, por sua vez, apresentou resultados melhores que o Bikmeans, competitivos com o EFD mas piores que

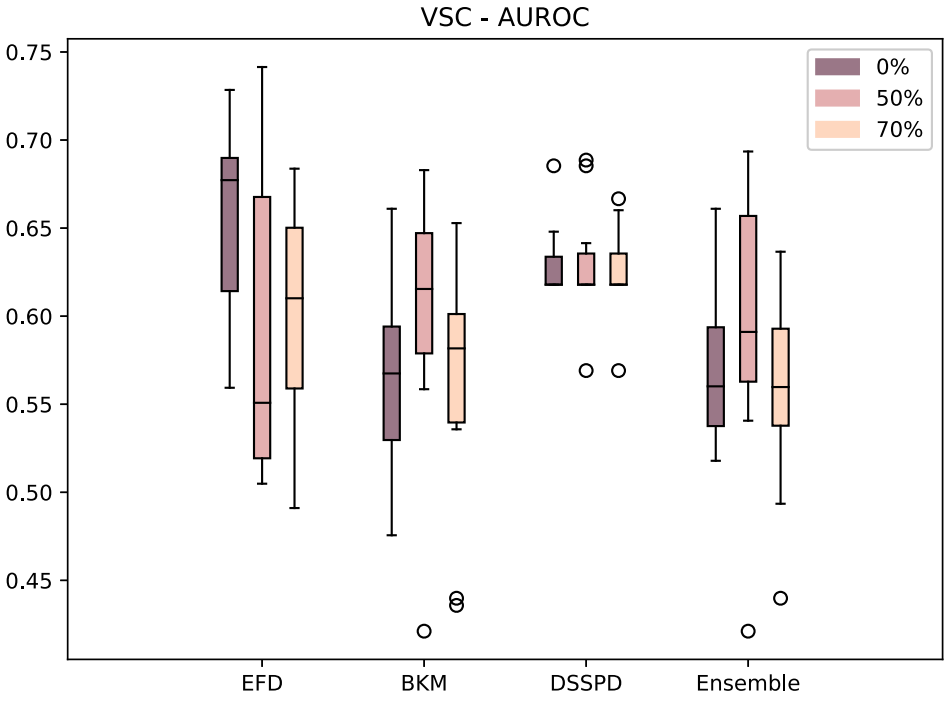
o DSSPD.

Figura 73 – Resultados para o problema VSC considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do VSC



(b) Resultados para AUROC do VSC



Fonte: Elaborado pelo autor (2024).

Já para o problema VSC, o EFD superou todos os métodos de discretização para

todas as taxas de *dropout*, tanto em AUPRC quanto em AUROC, exceto para 50% de *dropout* em AUROC, onde a mediana ficou inferior aos demais métodos. O comportamento do método *ensemble* aproximou-se do Bikmeans. O comportamento do DSSPD apresentou-se bem estável.

Por fim, para o problema HSC, o EFD apresenta resultados ligeiramente melhores que o Bikmeans. Para AUPRC, o DSSPD obteve o pior comportamento, para todas as taxas de *dropout*. O *ensemble* não superou a abordagem EFD. Já para AUROC, os resultados para todos os métodos de discretização ficaram mais próximos. Contudo, o EFD ainda obtém resultados ligeiramente melhores que os demais métodos. O Bikmeans supera o EFD no caso de 50% de *dropout* e o *ensemble* apresenta comportamento próximo ao DSSPD.

O método *ensemble* não foi capaz de fornecer resultados melhores que os métodos de discretização comparados. Uma possível explicação para o bom desempenho do DSSPD no problema mCAD pode estar associado à quantidade de *pseudotimes* envolvidos neste problema. Como é o problema com o menor número de *pseudotimes* (2), é possível que a unificação das redes parciais tenha gerado uma quantidade de relações regulatórias pequenas, de tal maneira que as relações tenham sido consideradas durante a avaliação das *top-k* relações. Contudo, investigações futuras ainda são necessárias.

4.6.3.4 Desempenho em Problemas Experimentais

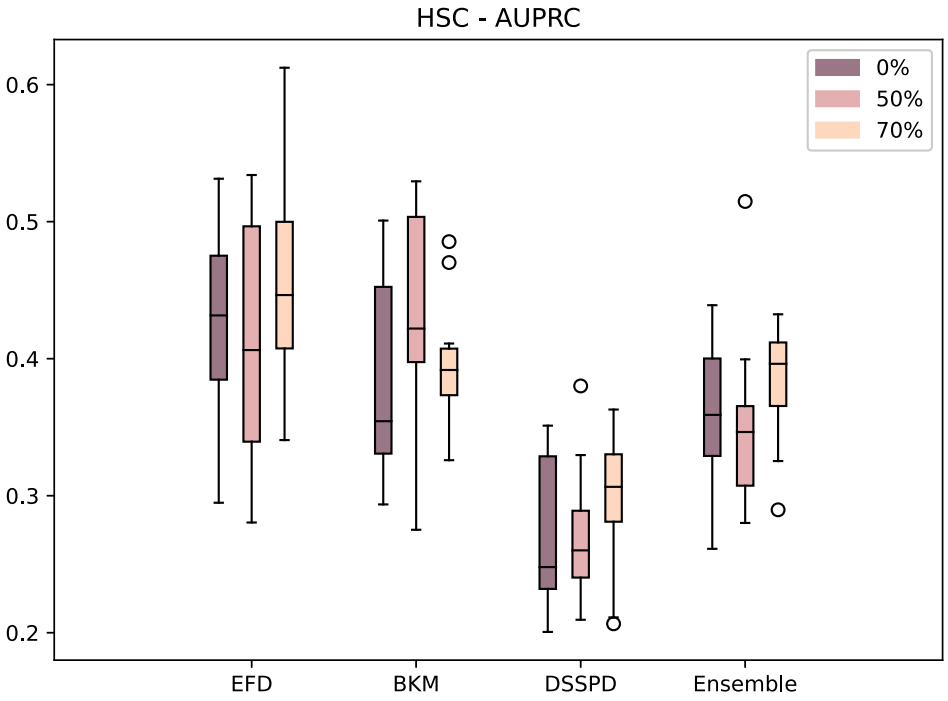
Os experimentos anteriores demonstraram que, dentre os métodos de discretização da literatura, o EFD apresenta o melhor desempenho. Além disso, o DSSPD é capaz de superar em diversas situações o Bikmeans e, em alguns casos, o EFD. Contudo, problemas experimentais não foram analisados quando considerado o EFD. Por esse motivo, experimentos computacionais foram realizados a fim de verificar o desempenho destes métodos de discretização quando aplicados a problemas experimentais. Para tal, foram considerados todos os problemas experimentais da configuração 500nTF.

Para todos os experimentos, o conjunto de funções da CGP é $\Gamma = \{$AND, OR, NOT, XOR, XNOR, NAND, NOR$\}$, $\mu = 1$, $\lambda = 4$, lb $= n_c$, and $n_r = 1$. Além disso, $n_c =$ 1500 e o número máximo de avaliações da função objetivo é de 2.000.000 e 5 execuções independentes. A métrica adotada é o EP, conforme sugerido em (PRATAPA *et al.*, 2020). Além disso, as três redes de referência foram consideradas.

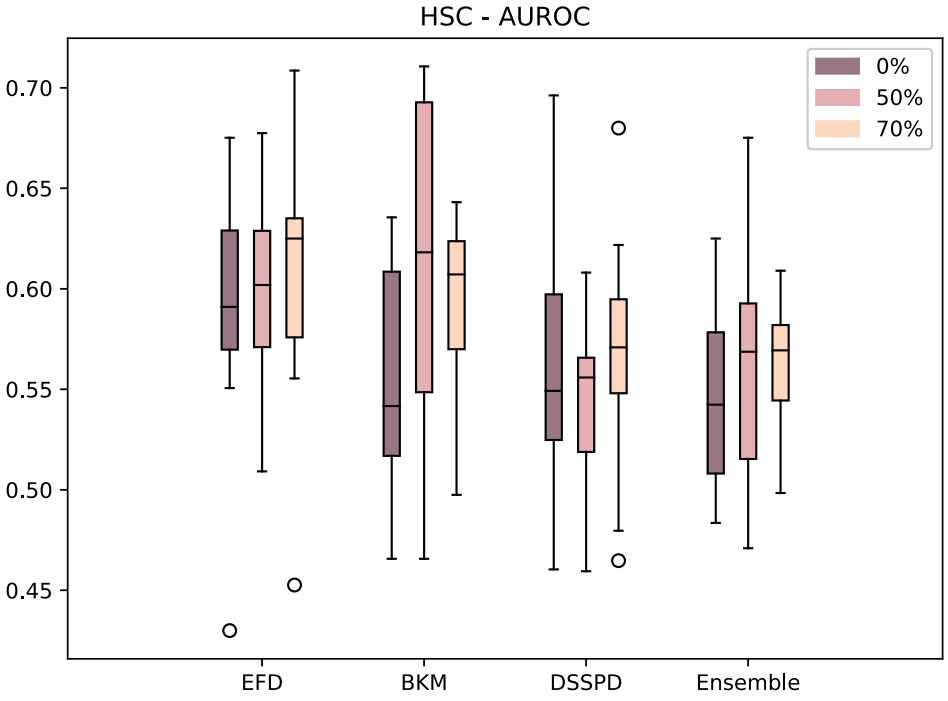
De acordo com a Figura 75, para a rede STRING, é possível perceber que o EFD não foi capaz de encontrar nenhuma relação regulatória correta para a maioria dos problemas, exceto para o problema mHSC-GM. Nesse caso, a mediana foi ligeiramente superior ao obtido pelo DSSPD mas no melhor caso o DSSPD ainda é melhor. O Bikmeans obteve resultados melhores em 4 problemas e o DSSPD nos outros três. A variabilidade dos resultados do Bikmeans é superior a todos os métodos de discretização considerados.

Figura 74 – Resultados para o problema HSC considerando AUPRC (a) e AUROC (b).

(a) Resultados para AUPRC do HSC



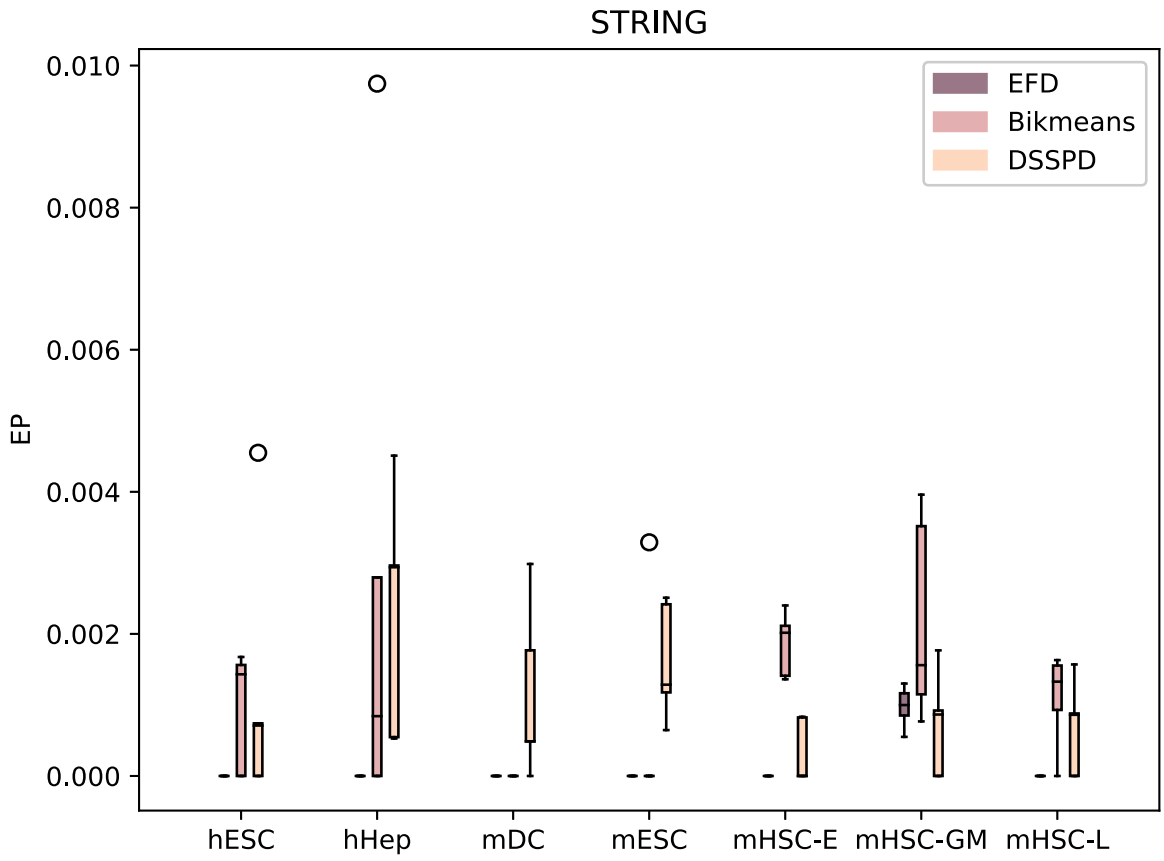
(b) Resultados para AUROC do mCAD



Fonte: Elaborado pelo autor (2024).

Já para a rede NonSpecific, os *boxplots* apresentados na Figura 76, os resultados variam substancialmente a depender do método de discretização. Novamente, exceto pelo

Figura 75 – *Boxplots* considerando o EP para todos os problemas e a rede de referência STRING.

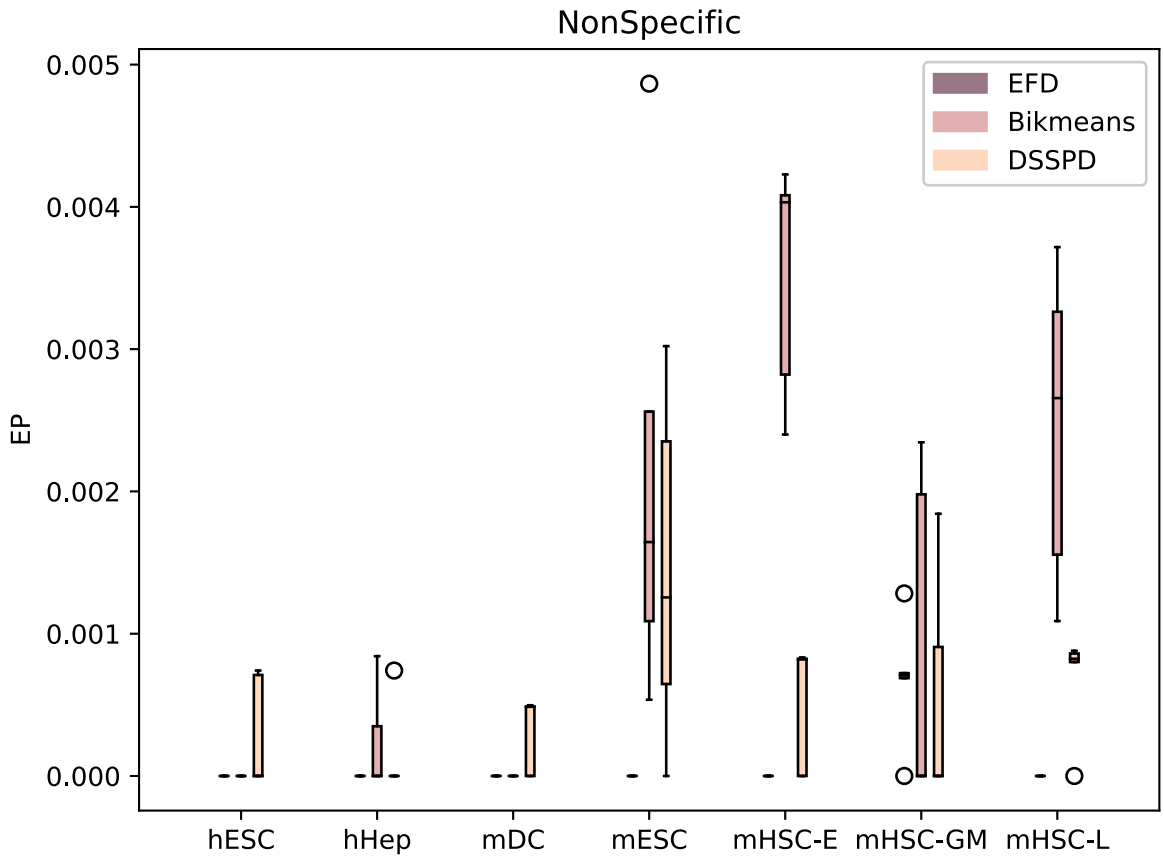


Fonte: Elaborado pelo autor (2024).

problema mHSC-GM, o EFD não foi capaz de encontrar relações regulatórias corretas para os demais problemas. O Bikmeans apresentou bons resultados principalmente nos problemas mESC e mHSC-L. Para os problemas hESC e mDC, somente o DSSPD foi capaz de encontrar relações regulatórias corretas. Para o problema mHSC-GM, a mediana dos resultados obtidos pelo Bikmeans e DSSPD são a mesma e a variabilidade do Bikmeans é maior.

Por fim, para a rede ChIP-Seq, conforme Figura 77, o EFD foi capaz de encontrar relações regulatórias em 3 problemas: mHSC-E, mHSC-GM e mHSC-L. Em nenhum destes problemas EFD obteve o melhor resultado, contudo foi competitivo com o DSSPD no problema mHSC-GM. O Bikmeans apresentou resultados melhores para os problemas hESC e hHep. Para o problema mDC as medianas entre DSSPD e Bikmeans são as mesmas e o DSSPD apresenta menor variabilidade. Vale destacar o bom desempenho obtido pelo DSSPD principalmente nos problemas mESC e mHSC-L com valores elevados de mediana.

Figura 76 – *Boxplots* considerando o EP para todos os problemas e a rede de referência NonSpecific.



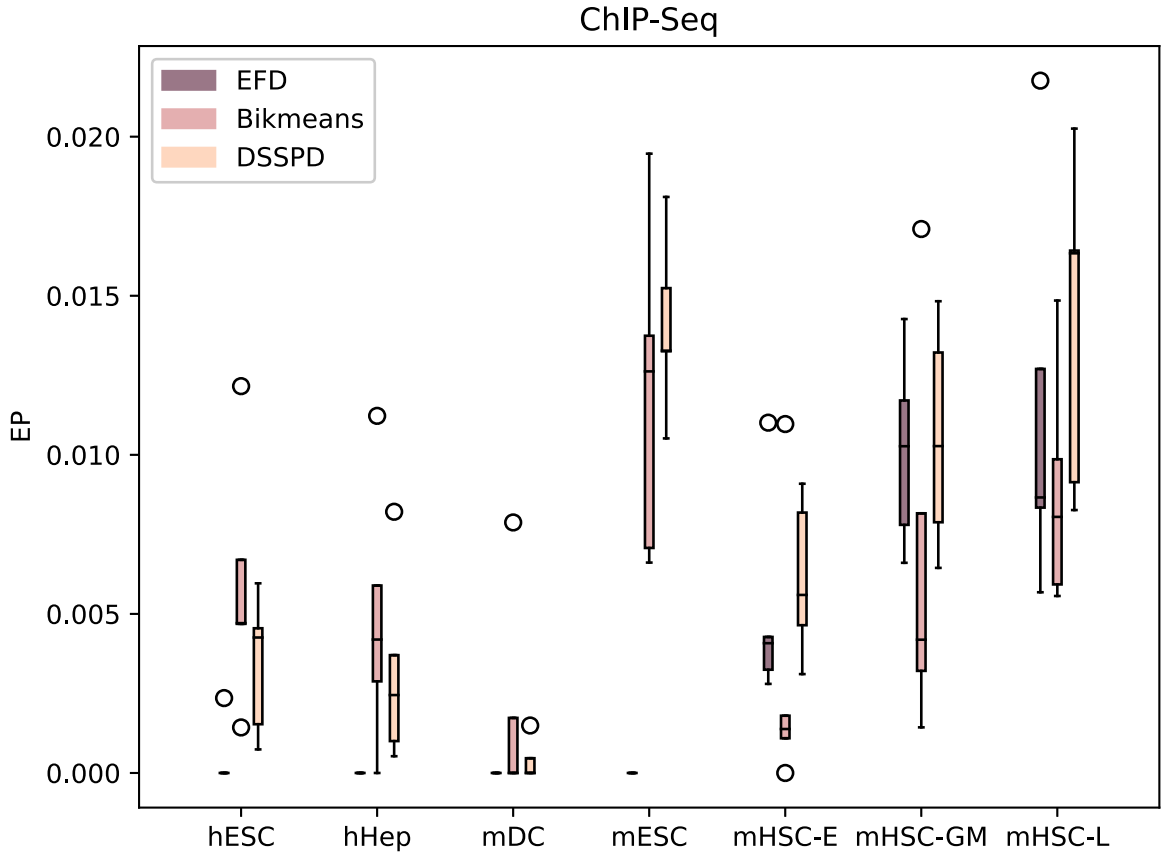
Fonte: Elaborado pelo autor (2024).

Baseado nestes *boxplots* e nas análises realizadas, conclui-se que apesar do EFD ter sido apontado como o melhor método de discretização para os problemas acurados, o mesmo não é observado para os problemas experimentais. É possível que os problemas sintéticos e acurados não estejam sendo capazes de representar as especificidades dos dados reais perfilados por scRNA-Seq. Uma discussão mais profunda sobre essas questões foi apresentada na Seção 2.13. Além disso, o DSSPD além de sempre apresentar resultados melhores que os obtidos pelo EFD, é capaz de superar o Bikmeans em quase todos os problemas com a rede de referência ChIP-Seq, e obtém resultados próximos, superando o Bikmeans em alguns problemas nas redes de referência STRING e NonSpecific.

4.6.4 Inferência do Modelo Booleano

Experimentos computacionais foram conduzidos para analisar o desempenho de operadores de mutação da CGP quando aplicados à inferência de GRNs usando problemas *benchmark* perfilados por scRNA-Seq na forma de séries temporais. Aqui, o objetivo é descobrir se a obtenção de soluções factíveis de maneira mais rápida tem um impacto

Figura 77 – *Boxplots* considerando o EP para todos os problemas e a rede de referência ChIP-Seq.



Fonte: Elaborado pelo autor (2024).

positivo na qualidade das soluções. Além disso, analisa-se a importância da otimização (redução do número de elementos lógicos). Os experimentos computacionais são compostos por duas partes: (i) o desempenho das variantes da CGP são comparados, e (ii) os resultados obtidos pelas duas melhores abordagens são comparadas com os resultados do algoritmo estado-da-arte para inferência de GRNs, GENIE3, por ser o melhor algoritmo para tal fim (PRATAPA *et al.*, 2020). Os problemas considerados aqui são da categoria acurados, exceto o problema GSD, como realizado nos experimentos de discretização.

Todos os métodos são implementados em C++ e os códigos estão disponíveis[12]. Os experimentos foram realizados em um computador com Ubuntu Server 20.04 LTS (HVM) com 16 vCPUs Intel (R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz e 30GB RAM. Os resultados são avaliados segundo o *framework* BEELINE (PRATAPA *et al.*, 2020).

[12] `hhttps://github.com/jeduaardo/bracis2021_cgp_mutation`

4.6.4.1 Análise Comparativa entre as Técnicas de CGP

Aqui compara-se a CGP tradicional com SAM e a CGP com SOMO em diferentes abordagens:

1. CGP: CGP padrão com SAM tanto para obter a solução factível quanto para otimizar,
2. SOMO: CGP com SOMO sem etapa de otimização,
3. SOMO-SAM: CGP com SOMO e etapa de otimização com SAM,
4. SOMO-SAM-R: CGP com SOMO e etapa de otimização com SAM e reinicialização da busca evolutiva,
5. SOMO-SAM-PQ50: CGP com SOMO e etapa de otimização com SAM e $p_q = 50\%$

Para a CGP padrão, foi utilizado $\lambda = 4$ e inicialização aleatória da população inicial. Quando considerando o SOMO, $\lambda = 1$, $p_f = 0$ e a inicialização da população sugerido em (HODAN; MRAZEK; VASICEK, 2020). Além disso, em (HODAN; MRAZEK; VASICEK, 2020) o valor de n_c é variável, considerando um número múltiplo do número de portas lógicas requeridos para implementar cada circuito. Contudo, a definição do número de portas lógicas para implementar um circuito não pode ser facilmente definido *a priori* e, portanto, fixamos $n_c = 100$, como usualmente considerando quando usando a CGP tradicional. Os demais parâmetros são $n_r = 1$ e $lb = n_c$. Para cada problema, 5 execuções independentes foram realizadas com um máximo de 100.000 avaliações da função objetivo. Para SOMO-SAM-R, o processo evolutivo é reiniciado com uma população diferente a cada 1.000 avaliações da função objetivo. Este valor foi determinado analisando a média de número de avaliações necessárias para encontrar uma solução factível. Além disso, o número de avaliações acumulado é usado como critério de parada.

As Tabelas 37 e 38 apresentam os resultados dos métodos de CGP, respectivamente, para AUPRC e AUROC. Os números ao lado do nome dos problemas representa a taxa de *dropout*. O melhor (máximo), primeiro quartil (Q1), mediana, média, terceiro quartil (Q3), e desvio padrão dos resultados são apresentados. Os melhores valores estão destacados em negrito. Além disso, métodos com asterisco possuem diferença estatística quando considerando o teste de Dunn para 95% de confiança. Os valores obtidos nos testes estão apresentados nas Tabelas 39 e 40 para AUPRC e AUROC, respectivamente.

Baseado nesses resultados, é possível perceber que os resultados de SOMO-SAM e SOMO-SAM-R são iguais em alguns casos. Isso é esperado, uma vez que a primeira solução factível foi obtida antes do primeiro ponto de reinicialização (1.000 avaliações da função objetivo). Em geral, a maior parte das abordagens de CGP testadas aqui, quando

Tabela 37 – Resultados de AUPRC para todos os problemas. O sufixo após o nome dos problemas é a taxa de *dropout*.

Algoritmo	**Melhor**	**Q1**	**Mediana**	**Média**	**Q3**	**Pior**	**DP**
			HSC-0				
CGP	0,4032	0,2535	0,2658	0,2901	0,3198	0,2241	5,63E-02
SOMO	**0,4637**	0,2691	0,2881	0,2986	0,3048	0,2253	6,01E-02
SOMO-SAM	0,4177	0,2679	0,2760	0,3023	0,3261	0,2396	**5,61E-02**
SOMO-SAM-R	0,4265	0,2634	0,2763	0,3044	0,3329	**0,2438**	5,74E-02
SOMO-SAM-PQ50	0,4626	**0,2960**	**0,3082**	**0,3277**	**0,3532**	0,2434	6,08E-02
			HSC-50				
CGP	0,4167	0,2998	0,3527	0,3440	0,3876	0,2509	5,28E-02
SOMO*	0,3738	0,3014	0,3093	0,3093	0,3282	0,2450	**3,46E-02**
SOMO-SAM	0,4290	0,3050	0,3771	0,3565	0,3872	**0,2718**	5,42E-02
SOMO-SAM-R	0,4325	0,2812	0,3569	0,3480	**0,4114**	0,2523	6,58E-02
SOMO-SAM-PQ50*	**0,4759**	**0,3275**	**0,3831**	**0,3660**	0,3952	0,2608	5,88E-02
			HSC-70				
CGP	0,3502	0,2714	0,2857	0,2915	0,2979	0,2462	**3,25E-02**
SOMO	**0,4490**	0,2682	0,2891	0,3074	**0,3419**	0,2284	6,41E-02
SOMO-SAM	0,3806	0,2608	0,2785	0,2962	0,3082	0,2562	4,39E-02
SOMO-SAM-R	0,4166	0,2751	0,2859	0,2964	0,3050	0,2417	4,42E-02
SOMO-SAM-PQ50	0,4001	**0,2891**	**0,3055**	**0,3151**	0,3361	**0,2597**	4,00E-02
			mCAD-0				
CGP	0,7508	0,5719	0,6452	0,6540	0,7508	0,5291	8,65E-02
SOMO	**0,8361**	0,5675	0,6049	0,6770	**0,8361**	0,5371	1,32E-01
SOMO-SAM	0,7844	**0,6238**	**0,6522**	**0,6871**	0,7844	**0,5843**	**8,22E-02**
SOMO-SAM-R	0,7844	**0,6238**	**0,6522**	**0,6871**	0,7844	**0,5843**	**8,22E-02**
SOMO-SAM-PF50	0,7631	0,5917	0,6369	0,6642	0,7631	0,5506	8,50E-02
			mCAD-50				
CGP*	0,6561	0,6081	0,6403	0,6281	0,6561	0,5515	**3,31E-02**
SOMO*	0,6020	0,5374	0,5737	0,5689	0,6020	0,5212	3,36E-02
SOMO-SAM*	0,6614	**0,5874**	**0,6536**	**0,6282**	**0,6614**	**0,5669**	3,85E-02
SOMO-SAM-R*	0,6614	**0,5874**	**0,6536**	**0,6282**	**0,6614**	**0,5669**	3,82E-02
SOMO-SAM-PF50	**0,6645**	0,5861	0,6431	0,6203	0,6431	0,5651	3,33E-02
			mCAD-70				
CGP	0,7624	0,5766	0,6452	0,6466	0,6926	0,5596	**7,13E-02**
SOMO	**0,8361**	0,5799	0,6073	0,6577	0,7407	0,5402	1,06E-01
SOMO-SAM	0,7960	**0,6274**	**0,6522**	**0,6793**	**0,7462**	**0,5843**	7,53E-02
SOMO-SAM-R	0,7960	**0,6274**	**0,6522**	**0,6793**	**0,7462**	**0,5843**	7,53E-02
SOMO-SAM-PF50	0,7747	0,6109	0,6369	0,6618	0,7284	0,5648	7,32E-02
			VSC-0				
CGP	0,4683	**0,2789**	0,3138	**0,3217**	0,3287	0,2338	6,67E-02
SOMO	0,3930	0,2660	0,2892	0,3011	0,3282	0,2205	5,23E-02
SOMO-SAM	0,3860	0,2643	0,3131	0,3134	**0,3634**	**0,2368**	**5,16E-02**
SOMO-SAM-R	0,4222	0,2456	0,2640	0,3021	0,3546	0,2293	6,91E-02
SOMO-SAM-PF50	**0,4951**	0,2749	**0,3286**	0,3290	0,3624	0,2287	7,41E-02
			VSC-50				
CGP	0,3730	0,2275	0,2590	0,2709	0,3071	0,1938	5,32E-02
SOMO	0,4035	**0,2665**	**0,2952**	**0,3024**	**0,3376**	**0,2218**	**5,10E-02**
SOMO-SAM	0,4645	0,2325	0,2541	0,2765	0,2696	0,2018	7,47E-02
SOMO-SAM-R	0,4317	0,2383	0,2634	0,2828	0,3144	0,2209	6,15E-02
SOMO-SAM-PF50	**0,4784**	0,2226	0,2675	0,2860	0,3147	0,2117	7,81E-02
			VSC-70				
CGP	**0,4671**	**0,3069**	0,3431	**0,3607**	**0,4395**	**0,2457**	7,50E-02
SOMO	0,4208	0,2479	0,2978	0,3007	0,3442	0,2235	6,69E-02
SOMO-SAM	0,4357	0,3014	**0,3674**	0,3496	0,3858	0,2409	6,10E-02
SOMO-SAM-R	0,4361	0,2596	0,3172	0,3213	0,3735	0,2191	6,94E-02
SOMO-SAM-PF50	0,4278	0,3013	0,3292	0,3289	0,3621	0,2329	**5,73E-02**

Fonte: SILVA *et al.* (2021).

Tabela 38 – Resultados de AUROC para todos os problemas. O sufixo após o nome dos problemas é a taxa de *dropout*.

Algoritmo	**Melhor**	**Q1**	**Mediana**	**Média**	**Q3**	**Pior**	**DP**
			HSC-0				
CGP	0,6376	0,5160	0,5413	0,5517	0,5951	0,4531	5,54E-02
SOMO	0,6186	0,5171	0,5598	0,5538	0,5739	0,5066	**3,78E-02**
SOMO-SAM	**0,7060**	0,5224	0,5654	0,5726	0,5967	0,4975	6,34E-02
SOMO-SAM-R	0,6969	0,5484	0,5568	0,5858	0,6108	**0,5103**	5,98E-02
SOMO-SAM-PQ50	0,6962	**0,5870**	**0,5960**	**0,6060**	**0,6467**	0,5089	5,71E-02
			HSC-50				
CGP	0,6827	0,5684	0,6197	0,6097	0,6465	0,5039	5,47E-02
SOMO	0,6625	0,5525	0,5834	0,5822	0,6207	0,4906	5,46E-02
SOMO-SAM	0,7109	0,5861	0,6268	0,6255	0,6542	**0,5412**	**5,12E-02**
SOMO-SAM-R	0,7141	0,5524	0,6006	0,6085	**0,6758**	0,5055	7,14E-02
SOMO-SAM-PQ50	**0,7596**	**0,6071**	**0,6493**	**0,6419**	0,6658	0,5391	5,64E-02
			HSC-70				
CGP	0,6172	0,5171	0,5488	0,5552	0,5912	0,4966	4,14E-02
SOMO	0,6564	0,5218	**0,5970**	0,5828	**0,6324**	0,5092	5,59E-02
SOMO-SAM	0,6516	0,5192	0,5616	0,5628	0,6161	0,4652	6,07E-02
SOMO-SAM-R	0,6777	0,5243	0,5604	0,5593	0,5705	0,5089	4,58E-02
SOMO-SAM-PQ50	**0,6861**	**0,5734**	0,5815	**0,5982**	0,6053	**0,5588**	**3,84E-02**
			mCAD-0				
CGP	0,6264	0,3750	0,5165	0,4978	0,6264	0,3242	1,22E-01
SOMO	0,6703	0,4148	0,4615	0,5176	0,6703	0,3681	1,28E-01
SOMO-SAM	**0,6923**	**0,4959**	**0,5330**	**0,5742**	**0,6923**	**0,4231**	**1,01E-01**
SOMO-SAM-R	**0,6923**	**0,4959**	**0,5330**	**0,5742**	**0,6923**	**0,4231**	**1,01E-01**
SOMO-SAM-PQ50	0,6484	0,4217	0,4973	0,5203	0,6484	0,3736	1,12E-02
			mCAD-50				
CGP	**0,5440**	0,4286	0,4863	0,4736	**0,5440**	0,3352	7,49E-02
SOMO*	0,4560	0,3626	0,4093	0,4055	0,4560	0,3352	5,11E-02
SOMO-SAM*	**0,5440**	**0,4341**	**0,4973**	**0,4852**	**0,5440**	**0,3791**	6,13E-02
SOMO-SAM-R*	**0,5440**	**0,4341**	**0,4973**	**0,4852**	**0,5440**	**0,3791**	6,13E-02
SOMO-SAM-PQ50	0,5055	0,4286	0,4918	0,4659	0,5055	0,3681	**4,66E-02**
			mCAD-70				
CGP	0,6319	0,4135	0,5055	0,4934	0,5412	0,3626	8,99E-02
SOMO	0,6703	0,4313	0,4753	0,5027	0,5632	0,3681	1,01E-01
SOMO-SAM	**0,6978**	**0,4973**	**0,5522**	**0,5604**	**0,6195**	**0,4231**	8,79E-02
SOMO-SAM-R	**0,6978**	**0,4973**	**0,5522**	**0,5604**	**0,6195**	**0,4231**	8,79E-02
SOMO-SAM-PQ50	0,6538	0,4602	0,500	0,5176	0,5797	0,3736	**8,75E-02**
			VSC-0				
CGP	0,6805	0,5014	0,5541	0,5608	0,6280	0,4211	7,78E-02
SOMO	0,6382	0,5006	**0,5581**	0,5440	0,5839	0,4236	**6,30E-02**
SOMO-SAM	0,6878	0,5089	0,5496	**0,5659**	0,6222	0,4707	7,06E-02
SOMO-SAM-R	0,6512	0,4848	0,5154	0,5462	**0,6394**	**0,4301**	8,36E-02
SOMO-SAM-PQ50	**0,7220**	**0,5167**	0,5508	0,5603	0,5982	0,4276	8,25E-02
			VSC-50				
CGP	0,6854	0,4280	0,5272	0,5113	0,5742	0,3415	1,01E-01
SOMO	0,6415	**0,4754**	**0,5488**	**0,5372**	**0,6085**	0,3699	8,32E-02
SOMO-SAM	0,7154	0,4467	0,4732	0,5049	0,5327	0,3659	1,01E-01
SOMO-SAM-R	0,6585	0,4591	0,5106	0,5217	0,5817	0,3894	**7,97E-02**
SOMO-SAM-PQ50	**0,7309**	0,4213	0,4699	0,5091	0,5602	**0,4114**	1,03E-02
			VSC-70				
CGP	**0,7707**	**0,5583**	0,6130	**0,6199**	**0,6929**	**0,4593**	9,13E-02
SOMO	0,7057	0,5071	0,5325	0,5480	0,5872	0,4317	**8,53E-02**
SOMO-SAM	0,7122	0,5630	**0,6276**	0,6023	0,6648	0,400	8,88E-02
SOMO-SAM-R	0,7415	0,5018	0,5780	0,5745	0,6433	0,4065	9,74E-02
SOMO-SAM-PQ50	0,7154	0,5545	0,5992	0,5841	0,6457	0,4057	9,95E-02

Fonte: SILVA *et al.* (2021).

Tabela 39 – Testes estatísticos considerando AUPRC. Valores representam o p-valor de Dunn e p_{kw} é o p-valor de Kruskal Wallis.

Problema	**SOMO-SAM**	**SOMO-SAM-PQ50**	**GENIE3**	p_{kw}
HSC-0	1,00E+00	3,10E-01 1,00E+00	3,10E-05 1,64E-03	7,99E-05
HSC-50	1,00E+00	7,61E-01 1,00E+00	1,82E-02 3,96E-02	3,68E-02
HSC-70	1,00E+00	4,31E-01 1,00E+00	3,10E-05 7,30E-04	5,57E-05
mCAD-0	1,00E+00	4,14E-01 1,00E+00	2,30E-05 6,28E-04	4,11E-05
mCAD-50	1,00E+00	3,85E-01 1,00E+00	2,80E-05 9,01E-04	5,71E-05
mCAD-70	1,00E+00	6,48E-01 1,00E+00	5,40E-05 3,42E-04	5,65E-05
VSC-0	1,00E+00	8,79E-01 1,00E+00	1,02E-04 1,89E-04	6,20E-05
VSC-50	1,00E+00	9,59E-01 1,00E+00	4,10E-03 3,49E-03	3,73E-03
VSC-70	1,00E+00	6,11E-01 1,00E+00	3,77E-04 4,80E-05	5,51E-05

Fonte: SILVA *et al.* (2021).

Tabela 40 – Testes estatísticos considerando AUROC. Valores representam o p-valor de Dunn e p_{kw} é o p-valor de Kruskal Wallis.

Problema	**SOMO-SAM**	**SOMO-SAM-PQ50**	**GENIE3**	p_{kw}
HSC-0	1,00E+00	3,10E-01 1,00E+00	3,10E-05 1,64E-03	7,99E-05
HSC-50	1,00E+00	7,61E-01 1,00E+00	1,82E-02 3,96E-02	3,68E-02
HSC-70	1,00E+00	4,31E-01 1,00E+00	3,10E-05 7,30E-04	5,57E-05
mCAD-0	1,00E+00	4,14E-01 1,00E+00	2,30E-05 6,28E-04	4,11E-05
mCAD-50	1,00E+00	3,85E-01 1,00E+00	2,80E-05 9,01E-04	5,71E-05
mCAD-70	1,00E+00	6,48E-01 1,00E+00	5,40E-05 3,42E-04	5,65E-05
VSC-0	1,00E+00	8,79E-01 1,00E+00	1,02E-04 1,89E-04	6,20E-05
VSC-50	1,00E+00	9,59E-01 1,00E+00	4,10E-03 3,49E-03	3,73E-03
VSC-70	1,00E+00	6,11E-01 1,00E+00	3,77E-04 4,80E-05	5,51E-05

Fonte: SILVA *et al.* (2021).

Tabela 41 – Contagem de algoritmos contando os melhores valores de mediana. Valores entre parênteses é a contagem dos algoritmos considerando a igualdade estatística.

Algorimo	**Contagem AUPRC**	**Contagem AUROC**
CGP	0(8)	0(8)
SOMO	1(8)	3(9)
SOMO-SAM	**4**(8)	**4**(9)
SOMO-SAM-R	3(8)	3(9)
SOMO-SAM-PQ50	**4**(8)	2(8)

Fonte: SILVA *et al.* (2021).

modelando GRNs, não apresentam diferença estatística quando comparadas entre si, tanto em AUPRC quanto em AUROC.

Foi contabilizado o número de vezes em que cada algoritmo alcançou o melhor resultado, em relação à mediana. Valores em parênteses consideram os testes estatísticos e, quando não existe diferença estatística, todos os métodos pontuam. Essa contagem é apresentada na Tabela 41. As abordagens que utilizaram uma etapa de otimização obtiveram resultados melhores que as demais. O esquema de reinicialização auxiliou na exploração do espaço de busca, principalmente na movimentação das soluções para fora de ótimos locais. A abordagem com reinicialização forneceu resultados melhores que a abordagem sem reinicialização. Contudo, não foi capaz de superar os resultados obtidos a partir da combinação SOMO-SAM. Além disso, quando utilizando $p_q = 50\%$, os resultados são melhores que os valores padrões apresentados em (HODAN; MRAZEK; VASICEK, 2020). A inicialização da população do SOMO é essencial para um bom desempenho do algoritmo, bem como o uso de $\lambda = 1$.

Por fim, baseado na Tabela 41, é possível concluir que os melhores métodos são SOMO-SAM e SOMO-SAM-PQ50 quando analisada a AUPRC. Para a AUROC, os melhores algoritmos são SOMO-SAM e SOMO. Contudo, aqui consideramos uma abordagem com o SOMO tradicional e SAM como etapa de otimização (SOMO-SAM) e outra abordagem variando o parâmetro p_q (SOMO-SAM-PQ50) para a comparação do próximo experimento.

4.6.4.2 Análise Comparativa com o GENIE3

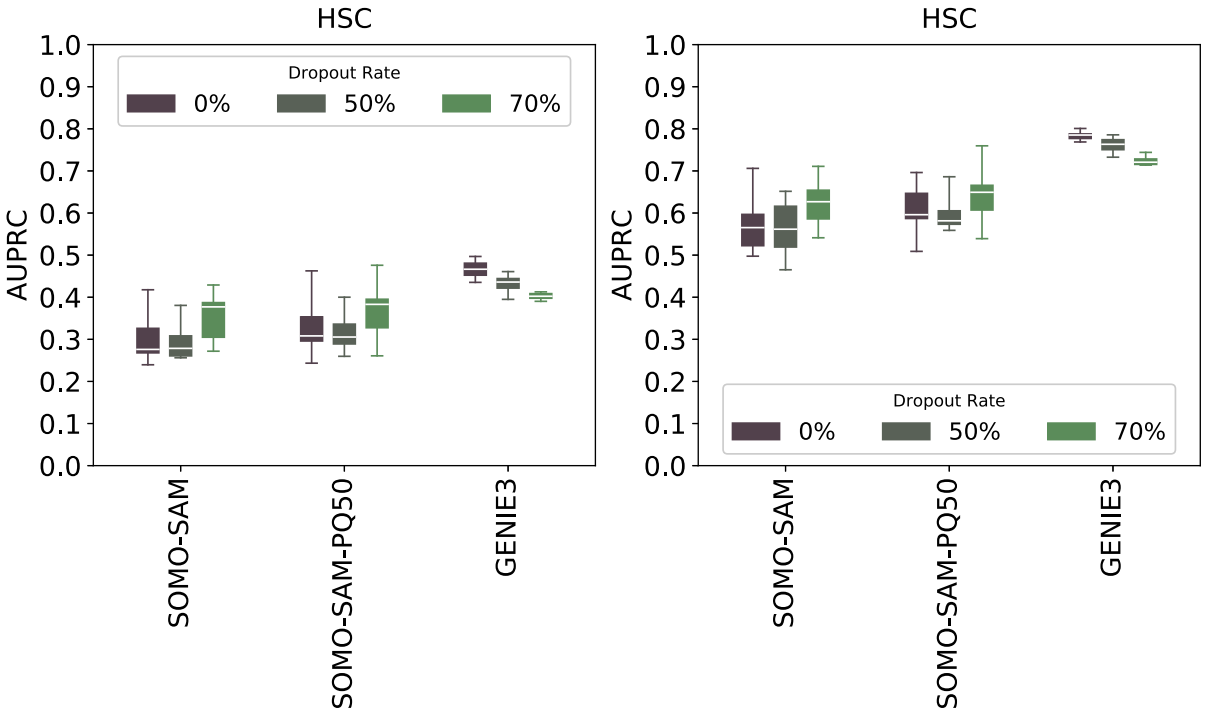
Como observado na seção anterior, as duas melhores variantes da CGP são SOMO-SAM e SOMO-SAM-PQ50. Essas duas variantes são agora comparadas com o GENIE3, apontado como o melhor algoritmo para a inferência de GRNs (PRATAPA *et al.*, 2020). Os resultados na forma de *boxplots* de BEELINE AUPRC e AUROC dos problemas HSC, mCAD e VSC são apresentados nas Figuras 78, 79 e 80, respectivamente.

Para as comparações, PPs foram utilizados e são apresentados na Figura 81. Os testes estatísticos de Kruskal-Wallis e Dunn estão disponíveis no material suplementar no

repositório.

Figura 78 – Resultados de AUPRC e AUROC para o problema HSC.

(a) AUPRC - HSC (b) AUROC - HSC



Fonte: SILVA *et al.* (2021).

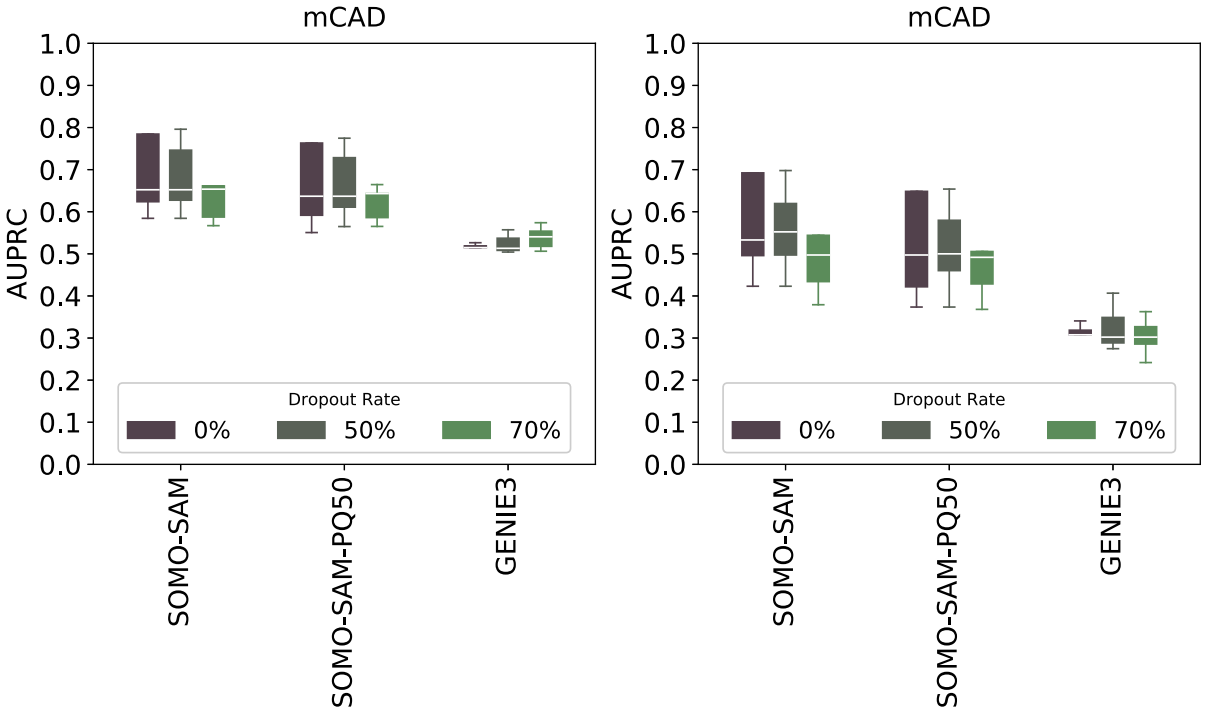
Baseado nos PPs apresentados na Figura 81 é possível concluir que, para AUPRC: (i) GENIE3 tem o melhor desempenho na maioria dos problemas (maior $\rho(1)$), seguido por SOMO-SAM e SOMO-SAM-PQ50, respectivamente, (ii) GENIE3 é a variante mais confiável (menor τ tal que $\rho(\tau) = 1$), seguido por SOMO-SAM-PQ50 e SOMO-SAM, respectivamente, e (iii) GENIE3 apresenta o melhor desempenho geral (maior área sob a curva), seguido por SOMO-SAM-PQ50 e SOMO-SAM, respectivamente. Portanto, GENIE3 é uma boa escolha quando considerado somente a AUPRC.

Por outro lado, quando considerado os PPs para AUROC, pode-se concluir que: (i) GENIE3 tem o melhor desempenho na maioria dos problemas, seguido por SOMO-SAM e SOMO-SAM-PQ50, respectivamente, (ii) SOMO-SAM e SOMO-SAM-PQ50 são as variantes mais confiáveis, e (iii) SOMO-SAM apresenta o melhor desempenho geral, seguido por SOMO-SAM-PQ50 e GENIE3, respectivamente. Logo, o SOMO-SAM destaca a importância do uso de uma etapa de otimização apropriada.

Além disso, o SOMO é bastante sensível ao parâmetro p_q, isto é, manter alguns nós inativos inalterados antes de aplicar o SOMO auxilia na obtenção de melhores resultados. Uma possível razão é o fato de que, quando utilizando SAM, adota-se $\lambda = 4$, então, CGP pode criar uma descendência mais diversa do que com $\lambda = 1$, como tradicionalmente

Figura 79 – Resultados de AUPRC e AUROC para o problema mCAD.

(a) BEELINE AUPRC - mCAD (b) BEELINE AUROC - mCAD



Fonte: SILVA *et al.* (2021).

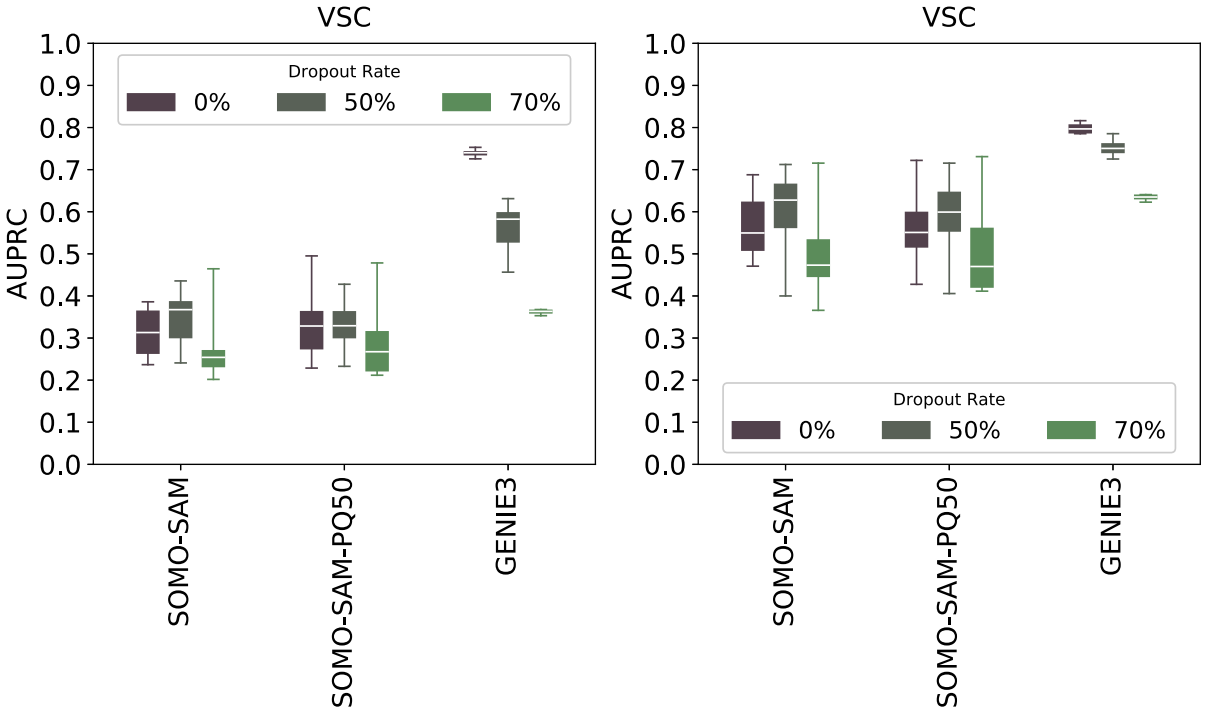
utilizado pelo SOMO.

Contudo, somente quando considerando o problema mCAD, abordagens que usam CGP obtiveram melhores resultados em AUPRC e AUROC. A respeito do mCAD é interessante destacar que esse problema é o único com 2 *pseudotimes*. O fluxo de avaliação considera somente as *top-k* relações regulatórias e, quando unificando as duas soluções parciais para obter a GRN final, é possível construir uma GRN menor (menor número de relações regulatórias). Como a avaliação considera apenas as relações regulatórias mais fortes, é possível que esse fato tenha levando a uma melhor avaliação desse problema.

Os testes estatísticos mostraram que não existe diferença estatística entre as abordagens que envolvem a CGP. Contudo, a diferença estatística é observada somente quando comparando as variantes de CGP com o GENIE3, reforçando a superioridade do GENIE3, como mostrado nos *boxplots*. Entretanto, é importante ressaltar que nestes experimentos, a discretização utilizada foi o Bikmeans. Como apresentado anteriormente, o Bikmeans pode não ser o método de discretização mais apropriado neste contexto. Dessa forma, experimentos adicionais ainda são necessários.

Figura 80 – Resultados de AUPRC e AUROC para o problema VSC.

(a) AUPRC - VSC (b) AUROC - VSC



Fonte: SILVA *et al.* (2021).

4.7 AVALIAÇÃO DO PROCESSO METODOLÓGICO

Experimentos computacionais foram realizados a fim de (i) analisar os métodos usados para a seleção de subconjuntos de genes, (ii) analisar o impacto da autorregulação, tanto na inferência quanto na avaliação, e (iii) analisar o impacto de diferentes métricas na avaliação de GRNs.

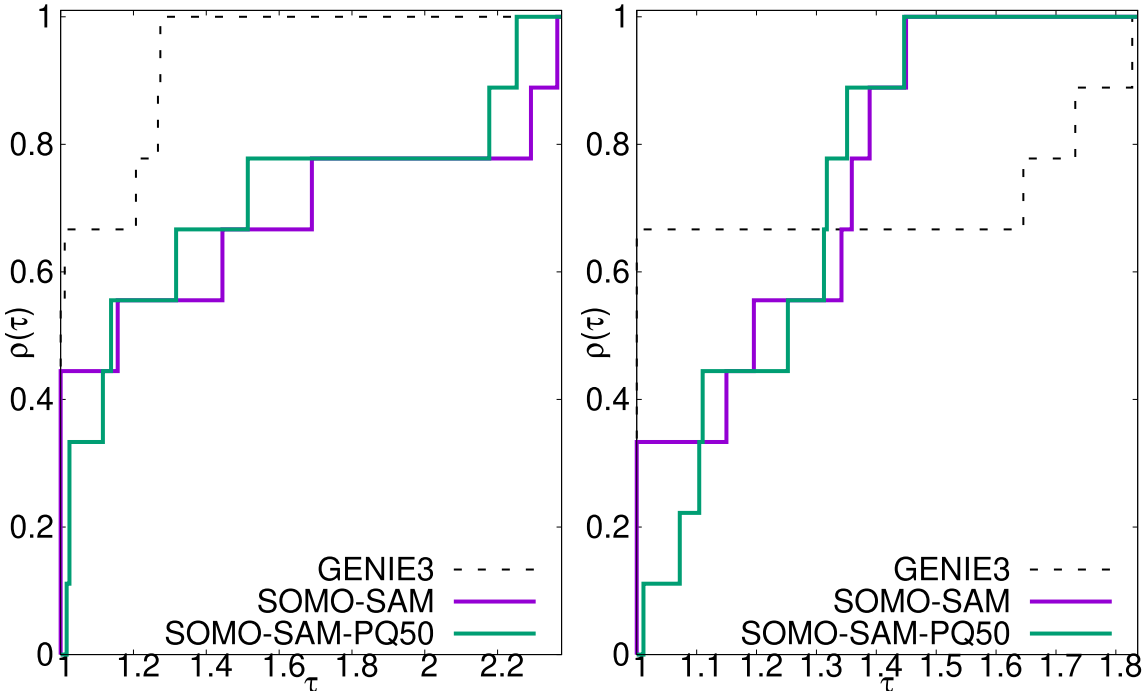
As análises incluem a investigação dos procedimentos comumente utilizados na literatura na seleção de subconjuntos de genes e os estudos que discutem o uso de diferentes métricas. As comparações apresentam as características que são levadas em consideração, ressaltando vantagens e desvantagens de cada uma das etapas.

Todos os problemas da categoria experimentais apresentados na Seção 4.1 foram considerados para análise, resumidos na Tabela 11. Também, foram utilizadas as configurações 500, 500+TF, 1000 e 1000+TF. Além disso, para comparação, foram considerados os algoritmos com melhor desempenho na avaliação de PRATAPA *et al.* (2020), a saber: GENIE3, GRNBOOST2, PIDC, PPCOR e SINCERITIES, além do CGPGRN. A Tabela 42 apresenta os algoritmos, os significados dos sufixos (quando aplicável), e as referências. Os parâmetros típicos para cada problema estão disponíveis no material suplementar no repositório. Por fim, consideram-se também todas as redes de referência previamente discutidas: STRING, *NonSpecific* e *ChIP-Seq*.

Figura 81 – PPs considerando a AUPRC e AUROC para as melhores abordagens.

(a) PPs para AUPRC. Áreas: SOMO-SAM (0,72), SOMO-SAM-PQ50 (0,76) e GENIE3 (1,00).

(b) PPs para AUROC. Áreas: SOMO-SAM (1,00), SOMO-SAM-PQ50 (0,98) e GENIE3 (0,94).



Fonte: SILVA *et al.* (2021).

Tabela 42 – Algoritmos considerados para os experimentos computacionais.

Algoritmo	Sufixos	Referências
GENIE3	-	(HUYNH-THU *et al.*, 2010)
GRNBOOST2	-	MOERMAN *et al.* (2019)
PIDC	-	CHAN; STUMPF; BABTIE (2017)
PPCOR	-	KIM (2015)
SINCERITIES	-	GAO *et al.* (2018)
CGP	DSSPD - discretização BKM - discretização NR - número de execuções por rede KM - k-means clustering	SILVA *et al.* (2020) SILVA *et al.* (2023) SILVA *et al.* (2024)

Fonte: Elaborado pelo autor (2024).

Os experimentos são divididos em três discussões principais: (i) a seleção de subconjuntos de genes, (ii) a consideração da autorregulação e (iii) as métricas e redes de referência.

4.7.1 Seleção de Subconjuntos de Genes

Os genes mais variantes são determinados usando o *general additive model* implementado no pacote "GAM" para R[13], conforme apresentado por PRATAPA *et al.* (2020) para computar a variância da expressão gênica. Dessa forma, o número de espécies apresentados para cada configuração dos conjuntos de dados variam quando consideram-se os TFs.

Inicialmente, analisamos os genes que foram selecionados e quantos destes genes possuem relações regulatórias em relação a cada uma das redes de referência. Essa informação é apresentada na Tabela 43.

De acordo com a Tabela 43, é possível perceber que, dentre os genes selecionados, o número de genes que possui relações regulatórias varia substancialmente a depender do conjunto de dados. Em alguns casos, como para o hESC, em relação à rede *NonSpecific*, apenas 12,2% dos genes selecionados possuem algum tipo de relação regulatória. Isso significa dizer que todos os demais genes (87,8%) estão presentes no *dataset*, podem e devem ser utilizados pelos algoritmos de inferência, mas não terão relações regulatórias válidas no momento de avaliação. Isso não só gera um desperdício de recursos computacionais durante o processo de inferência mas também ressalta que a seleção de genes adotada não foi representativa.

Em geral, as menores proporções de genes com relações regulatórias são apresentadas quando considera-se a rede STRING e as maiores proporções são apresentadas para a rede *ChIP-Seq*. Quando lida-se com dados scRNA-Seq, faz sentido de que redes específicas por célula ou tipo celular, como a *ChIP-Seq*, sejam mais representativas. Em relação à rede STRING, conforme ressaltado anteriormente, por sua natureza de representação de relações funcionais e não necessariamente regulações transcricionais, ajudam a explicar a baixa proporção de genes com relações regulatórias. As redes STRING e *NonSpecific* podem se tornar mais significativas quando utilizam-se dados de natureza multi-ômica, dada a natureza da construção de tais redes.

Além disso, é importante ressaltar que para as configurações 500 e 1000, sem incluir os fatores de transcrição, o número de espécies que são utilizadas pelos algoritmos devem ser, necessariamente, 500 e 1000, respectivamente. Contudo, quando os fatores de transcrição são incluídos, é esperado que para as configurações 500+TF e 1000+TF o número de espécies seja, no mínimo, 500 e 1000, respectivamente. Entretanto, isso não é observado para o problema mHSC-L nas configurações 1000 e 1000+TF, onde somente 692 espécies foram identificadas. Ao analisar os *datasets* disponibilizados por PRATAPA *et al.* (2020), percebe-se a presença de 4762 genes. Isso não justifica os valores de 692.

Isso é explicado através da maneira pela qual os *datasets* são gerados no *framework*

[13] `https://cran.r-project.org/web/packages/gam/index.html`

BEELINE. Existe um parâmetro padrão para realizar a correção Bonferroni (`-c`). Ao remover esse parâmetro, o número de espécies permanece como o esperado. Contudo, como objetivamos investigar o uso do *framework* exatamente como proposto em PRATAPA *et al.* (2020), e, ao remover o parâmetro `-c`, as configurações das redes diferem daqueles apresentados na publicação original, manteremos o uso da correção. Por sua vez, mesmo que deseje-se utilizar as configurações 1000 e 1000+TF para o problema mHSC-L, os *datasets* resultantes não conterão mais do que 692 espécies, a menos que o parâmetro `-c` seja removido.

Tabela 43 – Resumo da interseção dos genes selecionados com as redes de referência para todos os problemas e configurações. (#S número de espécies, GWR - genes com relações regukatórias, #R número de relações regulatórias).

			ChIP-Seq		NonSpecific		STRING	
	#S	Problema	GWR	#R	GWR	#R	GWR	#R
500nTF	500	hESC	310 (62%)	567	61 (12,2%)	60	92 (18,4%)	130
		hHep	378 (75,6%)	381	88 (17,6%)	102	161 (32,2%)	416
		mDC	34 (6,8%)	36	33 (6,6%)	34	87 (17,4%)	126
		mESC	367 (73,4%)	3359	206 (41,2%)	376	88 (17,6%)	251
		mHSC-E	459 (91,8%)	1372	169 (33,8%)	276	112 (22,4%)	288
		mHSC-GM	481 (96,2%)	2221	130 (26%)	197	89 (17,8%)	224
		mHSC-L	469 (93,8%)	3312	140 (28%)	218	62 (12,4%)	115
500TF	910	hESC	815 (89,56%)	4545	760 (83,52%)	3441	517 (56,81%)	4257
	948	hHep	874 (92,19%)	9939	832 (87,76%)	4129	656 (69,2%)	7523
	821	mDC	448 (54,57%)	756	643 (78,32%)	3067	487 (59,32%)	4815
	1120	mESC	977 (87,23%)	29613	896 (80,0%)	6893	648 (57,86%)	7762
	704	mHSC-E	691 (98,15%)	11557	447 (63,49%)	1425	300 (42,61%)	1371
	632	mHSC-GM	618 (97,78%)	7364	300 (47,47%)	742	206 (32,59%)	748
	560	mHSC-L	525 (93,75%)	4398	168 (30%)	279	74 (13,21%)	137
1000nTF	1000	hESC	753 (75,3%)	2131	500 (50%)	739	224 (22,4%)	425
		hHep	816 (81,6%)	2133	210 (21%)	297	342 (34,2%)	1113
		mDC	261 (26,1%)	310	272 (27,2%)	510	250 (25%)	849
		mESC	775 (77,5%)	11196	498 (49,8%)	1157	225 (22,5%)	706
		mHSC-E	968 (96,8%)	7151	361 (36,1%)	684	241 (24,1%)	741
		mHSC-GM	952 (95,2%)	8166	360 (36%)	719	273 (27,3%)	879
	692	mHSC-L	640 (92,49%)	5180	198 (28,61%)	317	86 (12,43%)	154
1000TF	1410	hESC	1260 (89,36%)	7084	1149 (81,49%)	4617	709 (50,28%)	5149
	1448	hHep	1331 (91,92%)	15558	1224 (84,53%)	5351	889 (61,4%)	9003
	1321	mDC	690 (52,23%)	1193	980 (74,19%)	3918	681 (51,55%)	5898
	1620	mESC	1385 (85,49%)	42795	1221 (75,37%)	8030	799 (49,32%)	8479
	1204	mHSC-E	1177 (97,76%)	21975	680 (56,48%)	1960	427 (35,47%)	1826
	1132	mHSC-GM	1089 (96,2%)	14135	531 (46,91%)	1357	357 (31,54%)	1311
	692	mHSC-L	640 (92,49%)	5180	198 (28,61%)	317	86 (12,43%)	154

Uma possibilidade para contornar o problema associado à presença de genes que não compartilham relações regulatórias é usar algoritmos de agrupamento, como o K-Means, para direcionar o algoritmo de inferência (ou busca). O uso de algoritmos de agrupamento para a identificação de genes altamente variantes foi discutido na Seção **2.13.1**. Por

esse motivo, conduzimos experimentos considerando o agrupamento dos genes, usando o K-Means, com número de *clusters* variando entre 2 e 10. O melhor valor para o número de *clusters* é determinado usando o coeficiente de silhueta, discutido na Seção 2.9.

Os resultados das proporções de genes que compartilham relações regulatórias para cada um dos problemas, em todas as configurações, considerando o uso de agrupamento, são apresentadas nas Tabelas 48, 49, 50, 51, 52, 53, e 54, disponíveis no Apêndice A. Três métricas são consideradas a fim de quantificar a qualidade do agrupamento: homogeneidade, completude e *V-Measure*, discutidas na Seção 2.9. De acordo com as tabelas, é possível perceber que:

- Para o problema hESC, a rede *ChIP-Seq* apresenta os melhores resultados de *V-Measure* em todas as configurações. Contudo, é importante ressaltar que para a configuração 500TF, todas as redes apresentam baixo valor de *V-Measure* (0.00478, 0.00783, and 0.30800, para STRING, *NonSpecific*, e *ChIP-Seq*, respectivamente). Além disso, para todos os casos, ao menos um dos *clusters* auxiliou na maximização da presença de genes com relações regulatórias compartilhadas.

- Para o problema hHep, a rede *ChIP-Seq* também apresenta os melhores resultados de *V-Measure* em todas as configurações. Para a rede STRING, em realação à rede *NonSpecific*, os resultados são melhores nas confgurações que não consideram fatores de transcrição. O oposto é observado quando os fatores de transcrição são considerados. Novamente, os valores de *V-Measure* para a configuração 500TF são baixos, exceto para a rede *ChIP-Seq* (0.09139, 0.25280, e 0.69538 para STRING, *NonSpecific*, e *ChIP-Seq*, respectivamente). É importante ressaltar que apenas para essa configuração o coeficiente de silhueta foi maximizado para um número de *clusters* (k) igual a 5, enquanto que as configurações 500nTF e 1000nTF possuem *k=2* e 1000TF com *k=3*. Por fim, para todos os casos, ao menos um dos *clusters* aumentou o número de genes com relações regulatórias em relação ao total. Quando considerando a configuração 500TF, em relação à rede STRING, observa-se que 90,09% dos genes no *cluster* 4 compartilham relações regulatórias. O mesmo é observado para a configuração 1000TF, com 91% no *cluster* 1.

- Para o problema mDC, a rede *ChIP-Seq* apresenta os melhores valores de *V-Measure* em todas as configurações. Quando considerando a rede *NonSpecific*, os resultados são melhores que aqueles obtidos para a rede STRING, exceto na configuração 500nTF. Além disso, o número de genes que compartilham relações regulatórias são substancialmente maiores nas configurações que consideram os fatores de transcrição. Exceto pela configuração 500TF, onde *k=4*, todos os demais casos utilizaram *k=2*, de acordo com o coeficiente de silhueta. Em todos os casos, ao menos um dos *clusters* aumentou a concentração de genes com relações regulatórias em relação ao total.

Quando considerando o *cluster* 0 da rede *NonSpecific*, 95,7% dos genes compartilham relações regulatórias (78,32% no total). O mesmo é observado para o *cluster* 0 da rede STRING (81,05% comparado a 59,32%), ambos na configuração 500TF.

- Para o problema mESC, os valores de *V-Measure* são melhores, em todas as configurações, quando considera-se a rede *ChIP-Seq*. A rede *NonSpecific* apresenta valores melhores que os apresentados na rede STRING, mas os valores são baixos, sendo menores ou iguais a 0,4641. Para a configuração 500nTF, a completude das redes *NonSpecific* e STRING são 1. Contudo, a homogeneidade é 0, para ambos os casos, levando o *V-Measure* também para 0. Em geral, para as configurações 500nTF e 500TF, o agrupamento não aumentou o número de genes que compartilham relações regulatórias. Contudo, existe melhoria para as configurações 1000nTF e 1000TF.

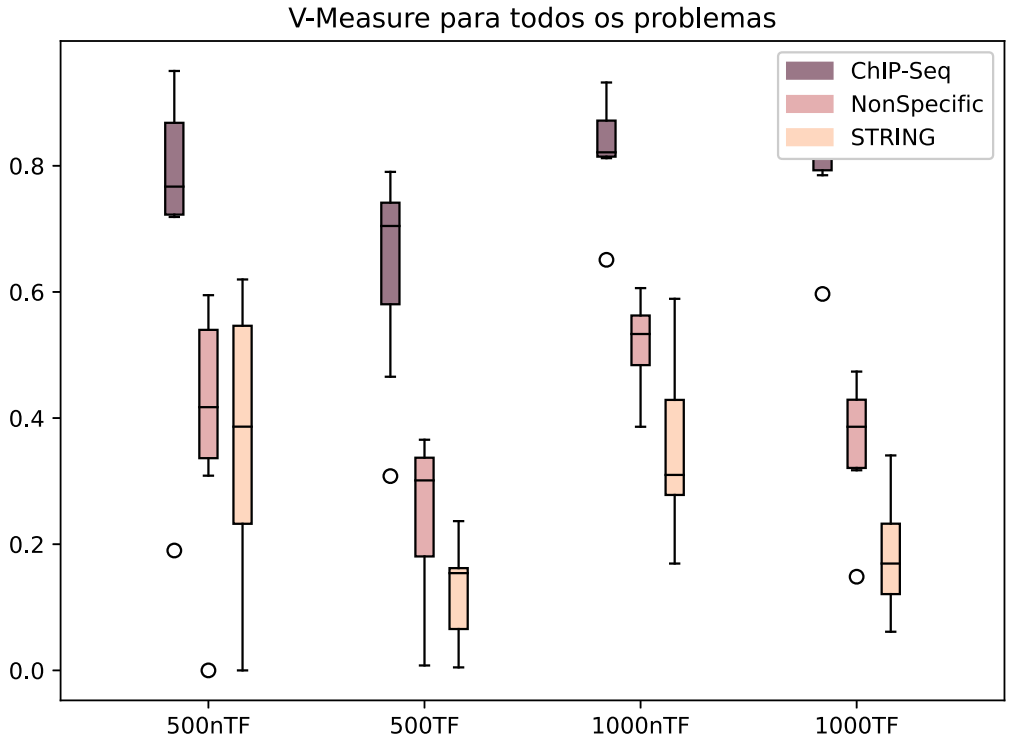
- Para o problema mHSC-E, novamente, os valores de *V-Measure* em todas as configurações são melhores quando considera-se a rede *ChIP-Seq*. Em geral, os resultados para a rede *NonSpecific* são melhores que aqueles obtidos para a rede STRING. Para todos os casos, ao menos um *cluster* aumentou o número de genes com relações regulatórias em relação ao total.

- Para o problema mHSC-GM, os valores de *V-Measure* são melhores quando considerando a rede *ChIP-Seq*. Os resultados para a rede *NonSpecific* são melhores que aqueles obtidos para a rede STRING, em todos os casos. Baseado na maximização do valor de coficiente de silhueta, o número de *clusters* varia de acordo com a configuração. O número de *clusters* é *k=2* para 500nTF, *k=4* para ambos 500TF e 1000nTF, e *k=3* para 1000TF. Para todos os casos, ao menos um *cluster* aumentou o número de genes que compartilham relações regulatórias.

- Para o problema mHSC-L, os melhores valores de *V-Measure* também são observados quando considera-se a rede *ChIP-Seq*. Os valores obtidos para a rede *NonSpecific* são melhores que aqueles obtidos para a rede STRING, em todos os casos. Contudo, os valores são baixos, sendo menores ou iguais a 0,38625. Exceto para a configuração 500nTF, onde *k=2*, todos os demais casos possuem o valor do coeficiente de silhueta maximizados quando *k=3*. Por fim, para todos os casos, ao menos um dos *clusters* aumentou o número de relações regulatórias.

Em resumo, quando consideram-se as redes *NonSpecific* e STRING, o valor de *V-Measure* não é maior que 0,62. Na maioria dos casos, esses valores são menores que 0,45 e 0,25 para as redes *NonSpecific* e STRING, respectivamente. Contudo, para a rede *ChIP-Seq*, os valores de *V-Measure* são melhores em todos os problemas e configurações, com resultados tipicamente maiores que 0,7 na maioria dos casos. Isso pode ser facilmente observado nos *boxplots* apresentados na Figura 82.

Figura 82 – *V-Measure* para todos os problemas e configurações.



Fonte: Elaborado pelo autor (2024).

Além disso, como será discutido nas Seções **4.7.2**, e **4.7.3**, o agrupamento auxilia na obtenção de melhores valores para AUPRC e AUORC quando considerada a rede *ChIP-Seq* para o CGP-DSSPD e o CGP-KM-DSSPD.

Uma segunda análise pode ser realizada considerando os subconjuntos de genes disponibilizados nos trabalhos que apresentam os problemas hHep e hESC (CAMP *et al.*, 2017; CHU *et al.*, 2016). Cruzamos esses genes com os *datasets* disponibilizados por PRATAPA *et al.* (2020) e intersectamos com as redes de referência. Para o problema hHep, é importante ressaltar que dos 379 genes apresentados em (CAMP *et al.*, 2017), aproximadamente 94% foram encontrados nos dados brutos apresentados em PRATAPA *et al.* (2020). Já para o problema hESC, três subconjuntos são apresentados, contendo 150, 2178 e 3247 genes (CHU *et al.*, 2016). Para os conjuntos com 150 e 3247 genes, todos os genes estão presentes nos dados disponibilizados em (PRATAPA *et al.*, 2020). Contudo, para o conjunto de 2178, cerca de 94% dos genes estavam presentes. Logo, o conjunto de dados disponibilizados por (PRATAPA *et al.*, 2020) para o problema hESC está incompleto.

De maneira similar ao realizado anteriormente, a quantidade desses genes que compartilham relações regulatórias também pode ser analisado. Essa informação é apresentada na Tabela 44.

Tabela 44 – Resumo da interseção dos genes selecionados com as redes de referência considerando suas publicações originais (#S número de espécies, GWR - genes com relações regulatórias, #R número de relações).

		ChIP-Seq		NonSpecific		STRING	
Problema	#S	GWR	#R	GWR	#R	GWR	#R
hHep	355	234 (65,92%)	327	128 (36,06%)	163	78 (21,97%)	252
hESC	150	138 (92%)	479	73 (48,67%)	153	93 (62%)	508
	2036	1569 (77,06%)	44271	1920 (94,30%)	25412	1962 (96,37%)	37504
	3247	1202 (37,02%)	19966	1904 (58,64%)	4870	2666 (82,11%)	10202

Fonte: Elaborado pelo autor (2024).

É possível comparar os resultados apresentados na Tabela 44 com aqueles apresentados na Tabela 43, considerando o hHep (355 genes) e o hESC (150 genes), em relação à configuração 500nTF. Ao considerar o problema hHep, a proporção de genes com relações regulatórias (#GWR) permanece similar, com variações em torno de 10%. Contudo, o número de relações regulatórias (#R) é menor para as redes *NonSpecific* e STRING considerando o conjunto de genes selecionados pelos experimentalistas. Para o problema hESC, a proporção de genes com relações regulatórias é maior quando considera-se o conjunto de genes selecionados pelos experimentalistas. Por exemplo, em relação à rede *ChIP-Seq*, 92% dos genes compartilham relações regulatórias em oposição à 62% quando considerando os genes selecionados pelo GAM em PRATAPA *et al.* (2020). Isso é um indicativo de que o subconjunto de genes selecionados pelos experimentalistas é mais relevante que aqueles selecionados pelo GAM.

Agora, assumindo que os 355 genes do problema hHep apresentado em (CAMP *et al.*, 2017) e os conjuntos de 150, 2036 e 3247 genes para o problema hESC (CHU *et al.*, 2016) são os mais interessantes a serem analisados, tais valores podem ser utilizados como *threshold* para gerar conjuntos de genes mais significativos da mesma maneira que realizado por PRATAPA *et al.* (2020). Então, podemos cruzar os dois conjuntos e determinar quantos genes foram selecionados igualmente por ambos os algoritmos de seleção de características. Esta comparação é apresentada na Tabela 45.

Tabela 45 – Interseção entre os genes apresentados nas publicações originais e os subconjuntos gerados pelo GAM.

	Referência	GAM	Proporção
hHep	355	106	29,86%
hESC	150	3	2%
	2.036	221	10,85%
	3.247	1.332	41,02%

Como pode ser visto na Tabela 45, apenas 3 de 150 genes (2%) foram encontrados

para o problema hHep. Esse número não é maior que 41% para todos os casos. Isso ressalta que o critério utilizado pelo algoritmo de seleção pode diferir significativamente daqueles usados pelos experimentalistas, resultando em conjuntos diferentes. Além disso, a literatura ressalta que a variância não pode ser usada como um indicador direto de HVGs YIP; SHAM; WANG (2019) por causa da heterocedasticidade presente nos dados de expressão gênica. Dessa forma, o uso do GAM enquanto algoritmo de seleção de características é questionável.

4.7.2 *Motifs* de Rede

Como ressaltado anteriormente, a autorregulação é um *motif* de rede extremamente importante para a manutenção de vida em diversos organismos. Contudo, é comum que os nós de autorregulação sejam eliminados das redes de inferência no momento da avaliação. Experimentos computacionais foram realizados, considerando todos os problemas, configurações e redes de referência, para avaliar as diferenças entre os resultados obtidos levando-se em consideração a presença ou ausência da autorregulação.

Aqui, já consideramos as diferentes possíveis métricas de avaliação e as medidas básicas da matriz de confusão. A Tabela 46 apresenta os resultados comparativos das diferenças relativas no desempenho. A referência é não considerar a autorregulação. O valor apresentado na coluna "Min" indica que não utilizar a autorregulação apresenta melhores resultados. Por outro lado, o valor apresentado na coluna "Max" indica que considerar a autorregulação apresenta melhores resultados.

De acordo com a Tabela 46, no que concerne à configuração 500nTF, é possível perceber que para os problemas hESC, mHSC-E, mHSC-GM e mHSC-L, existe grande variabilidade quando considera-se ou não a autorregulação. Contudo, na mediana, a diferença é zero. Os maiores desvio padrão são observados para os problemas hESC e mHSC-GM.

Em uma análise mais detalhada para o problema hESC-500nTF, por rede de referência, é possível perceber que essa ampla diferença é principalmente gerada por um *outlier* na rede *NonSpecific*, como apresentado na Figura 83a. De maneira geral, o comportamento por rede de referência apresenta maior variação quando considera-se a rede *ChIP-Seq*.

A mesma análise pode ser realizada para os problemas mHSC-E, mHSC-GM e mHSC-L, coforme apresentado nas Figuras 83b, 83c, e 83d, onde é possível observar que: (i) para mHSC-E, existe grande variabilidade para todas as redes, mas essas diferenças estão entre -20% e 20%; (ii) para mHSC-GM, o outlier de -1141,67% acontece no MCC para o algoritmo PPCOR (ver material suplementar) e representa a diferença, em valores absolutos, de 0,00012 (sem autorregulação) e -0,00125 (com autorregulação); como os valores são muito baixos, suas diferenças percentuais tornam-se muito grandes; e (iii) para

Tabela 46 – Comparação do desempenho relativo para todas as configurações métricas e redes considerando ou não a autorregulação. A referência é sem autorregulação. Resultados estão apresentados como percentual das diferenças relativas.

		Min	Q1	Média	Mediana	Q3	Std	Max
500nTF	hESC	-12,71	0	2,37	0	0	35,30	783,93
	hHep	-23,48	-0,17	-0,48	0	0	2,84	14,81
	mDC	-96,98	0	-0,55	0	0	7,05	4,44
	mESC	-44,12	-0,34	-0,91	0	0	5,33	37,5
	mHSC-E	-104,72	0	-1,44	0	0	13,33	87,27
	mHSC-GM	-1141,67	0	-3	0	0	48,88	153,81
	mHSC-L	-111,71	0	0,24	0	0	17,64	334,78
500TF	hESC	-100	0	-2,3	0	0	15,70	15,62
	hHep	-36,36	-0,24	0,032	0	0	8,91	200
	mDC	-100	0	-2,04	0	0,17	16,03	66,67
	mESC	-100	0	-2,09	0	0,14	18,33	200
	mHSC-E	-86,96	0	0,16	0	0	9,64	90,33
	mHSC-GM	-72,58	0	-0,11	0	0	6,05	52,61
	mHSC-L	-75,36	0	-0,15	0	0	7,88	100
1000nTF	hESC	-33,33	0	0,03	0	0	1,50	4,95
	hHep	-133,33	-0,12	-1,38	0	0	10,77	33,33
	mDC	-29,96	-0,11	-0,96	0	0	5,05	50
	mESC	-130,19	-0,19	-1,25	0	0	8,75	75
	mHSC-E	-188,79	0	13,01	0	0	344,96	8450
	mHSC-GM	-26,29	-0,13	-0,47	0	0	4,45	60
	mHSC-L	-357,89	0	0,58	0	0	21,61	349,36
1000TF	hESC	-33,33	-0,10	-0,09	0	0	3,98	50
	hHep	-22,22	-0,24	-0,42	0	0	1,98	8,33
	mDC	-11,93	-0,12	-0,48	0	0	1,82	5
	mESC	-50	-0,12	-0,23	0	0	2,59	19,05
	mHSC-E	-105,44	-0,15	-0,84	0	0	9,06	100
	mHSC-GM	-1072,73	-0,16	-2,83	0	0	45,16	126,32
	mHSC-L	-357,89	0	0,45	0	0	21,61	349,35

Fonte: Elaborado pelo autor (2024).

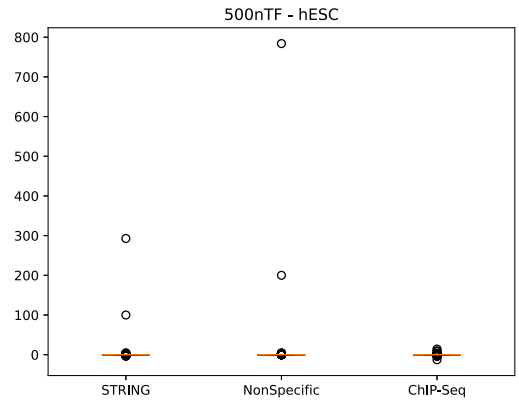
mHSC-L, a variabilidade é baixa na rede STRING, e os *outliers* são majoritariamente observados naqueles casos em que existem melhorias nos resultados quando considerada a autorregulação.

Expandindo as análises para as métricas, mas considerando-se todas as redes e problemas, os *boxplots* apresentados na Figura 84 são obtidos.

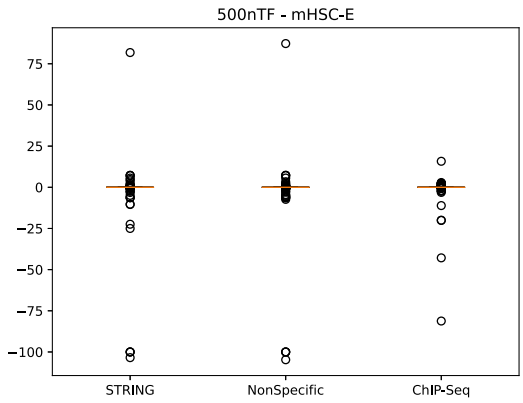
De acordo com a Figura 84 é possível perceber que grandes variações acontecem

Figura 83 – Resultados para os problemas considerando todas as métricas para as redes. A referência é sem autorregulação.

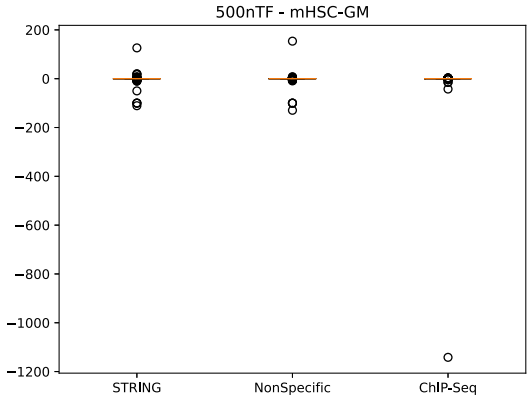
(a) Resultados para 500nTF - hESC



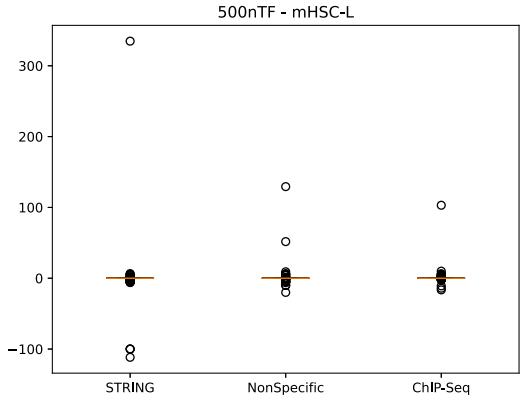
(b) Resultados para 500nTF - mHSC-E



(c) Resultados para 500nTF - mHSC-GM



(d) Resultados para 500nTF - mHSC-L

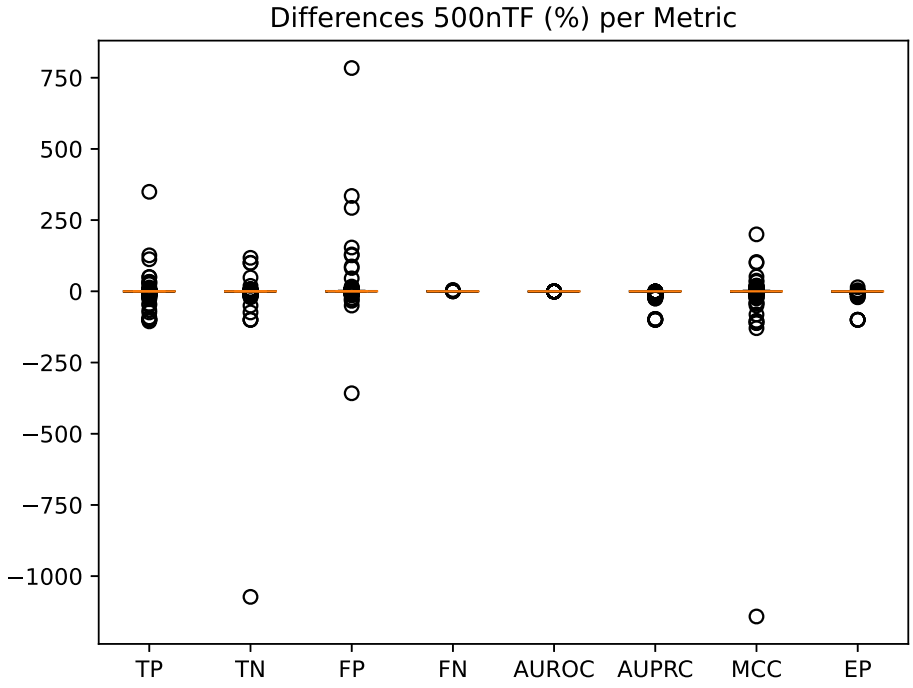


Fonte: Elaborado pelo autor (2024).

em TP, TN, FP, e MCC. Em relação ao TP, os resultados tornam-se piores quando considera-se a autorregulação. Contudo, para as demais métricas, especialmente FP, existe uma melhoria quando a autorregulação é considerada. Isso pode acontecer devido ao fato de que o fenômeno biológico modelado pelos problemas considerados aqui possuem poucas autorregulações. Desta forma, quando a autorregulação é obtida pelo algoritmo de inferência, há variação no número de TP.

Quando os problemas da configuração 500TF são analisados, os valores mínimos não são menores que 100% e os maiores valores não ultrapassam os 200%. Essa grande faixa de diferença é observada para o problema mESC. A Figura 85 torna clara que esses pontos referem-se à rede STRING e que constituem *outliers*. Um ponto importante a ressaltado aqui é que esses valores exatos (100% e 200%) são geralmente associados ao TP, TN, FP e FN. De maneira geral, esses valores são baixos (material suplementar). Por exemplo,

Figura 84 – Comparação por métrica para todos os problemas e redes de referência. A referência é sem autorregulação.



Fonte: Elaborado pelo autor (2024).

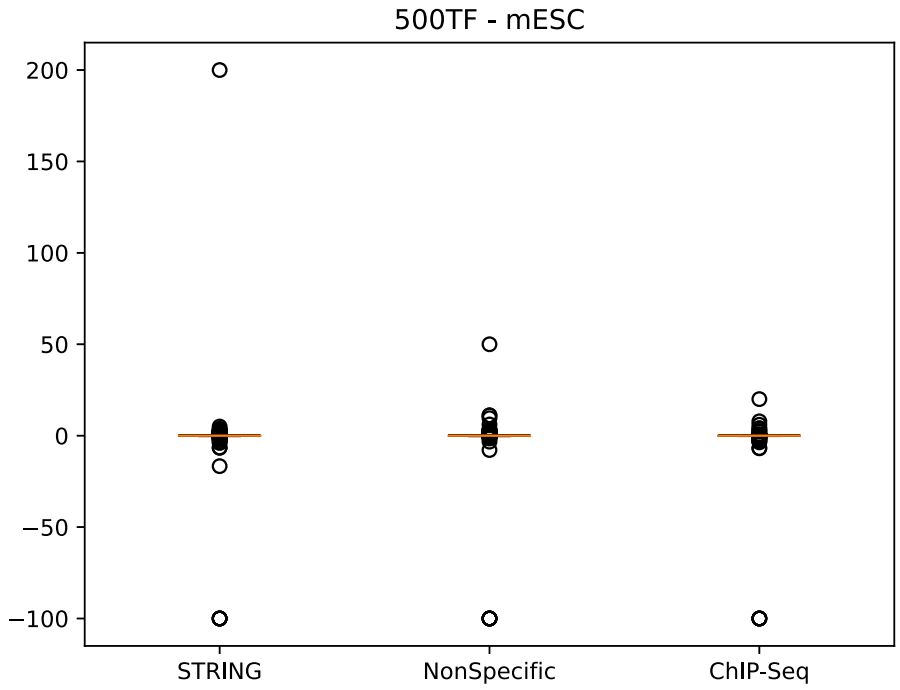
em alguns casos, o número de TP é 1. Então, ao realizar uma nova inferência/avaliação, pode ser que não hajam TPs. Isso causa uma diferença de -100%. Esses resultados são apresentados na Figura 86.

Continuando a análise, para 1000nTF, as diferenças são significativas para os problemas mHSC-E e mHSC-L, estando entre -188,79% e 8450% para o primeiro e -357,89% e 349,35% para o segundo. Os *boxplots* dessas diferenças, por rede, são apresentadas nas Figuras 87a e 87b para os problemas mHSC-E e mHSC-L, respectivamente.

De acordo com a Figura 87a, é possível perceber que os *outliers* ocorrem na rede *ChIP-Seq*. A respeito do valor máximo de 8450%, isso é observado para o algoritmo PPCOR. Ao analisar os resultados do PPCOR com e sem autorregulação é claro que a diferença também ocorre para o MCC, e são valores baixos (-2×10^{-5} sem autorregulação e $-1{,}7\times10^{-3}$ com autorregulação). Para a rede STRING, existe pequena variação, assim como para *NonSpecific*.

Para o problema mHSC-L, na Figura 87b, o valor mínimo acontece na rede *ChIP-Seq* e o valor máximo ocorre na STRING. Novamente, tanto as redes STRING quanto a *NonSpecific* apresentaram baixa variabilidade. Aprofundando a análise, a Figura 88 apresentam as diferenças por métrica, onde pode ser visto que as diferenças de 8450% ocorre em FP e MCC. Uma vez que MCC leva em consideração FP, essa repetição do

Figura 85 – Comparação de rede das diferenças entre as métricas para o problema mESC na configuração 500TF A referência é sem autorregulação.



Fonte: Elaborado pelo autor (2024).

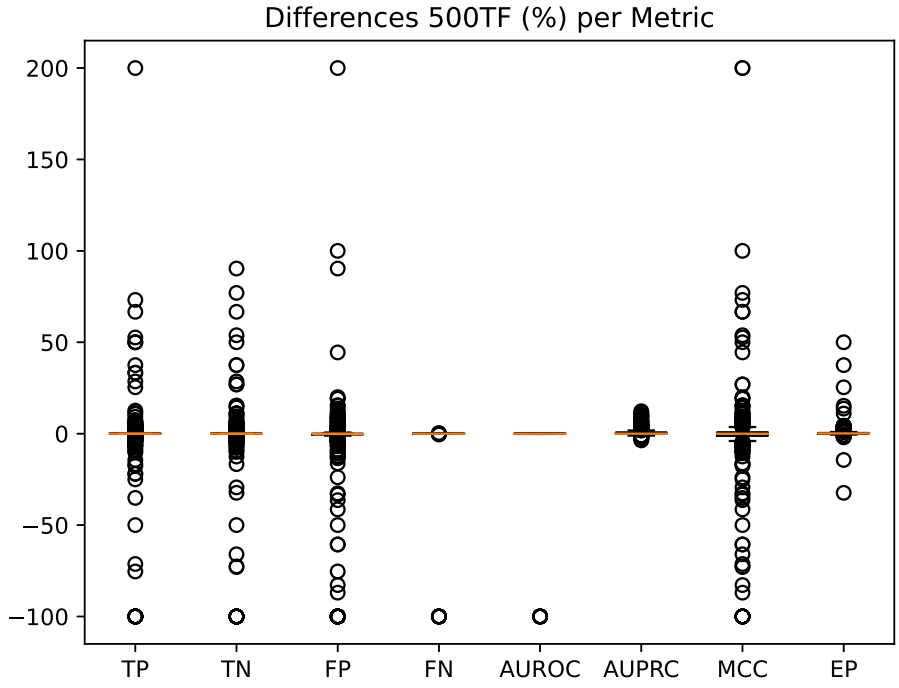
outlier é esperada.

Por fim, quando considerada a configuração 1000TF, as maiores diferenças são observadas nos problemas mHSC-GM e mHSC-L. Os *boxplots* das Figuras 89a e 89b mostram que:

- para mHSC-GM, o *outlier* acontece na rede *NonSpecific*. Novamente, essa diferença é observada no MCC devido aos baixos valores obtidos na avaliação dessa métrica. A rede *ChIP-Seq* apresenta baixa variabilidade.
- para mHSC-L, os *outliers* acontecem na rede STRING, onde considerar a autrregulação é melhor, e na *ChIP-Seq*, onde considerar a autorregulação é pior.

Os *boxplots* para todos os problemas, considerando tanto as análises por rede e por métrica estão disponíveis no material suplementar. Em geral, as maiores diferenças dos resultados com autorregulação e sem autorregulação são observados quando considerado o MCC. Contudo, é importante ressaltar que, na média, não existe diferença entre considerar ou não esse *motif* de rede. Uma possível explicação para isso reside no fato de que o número de vezes em que autorregulações ocorrem nos *datasets* utilizados é baixo e, na maioria dos casos, os algoritmos não foram capazes de identificar tal *motif* de rede. Por esse motivo,

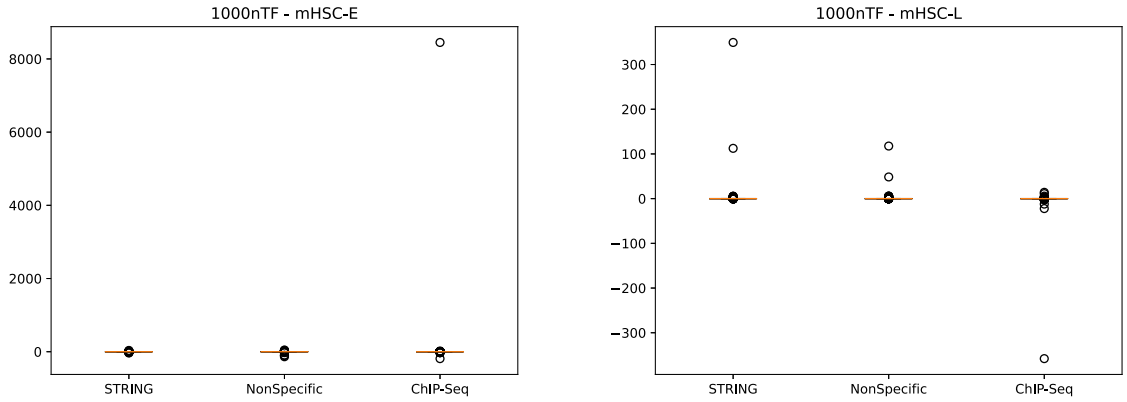
Figura 86 – Comparação de métricas das diferenças para o problema mESC na configuração 500TF. A referência é sem autorregulação.



Fonte: Elaborado pelo autor (2024).

Figura 87 – Resultados para os problemas considerando todas as métricas e redes. A referência é sem autorregulação.

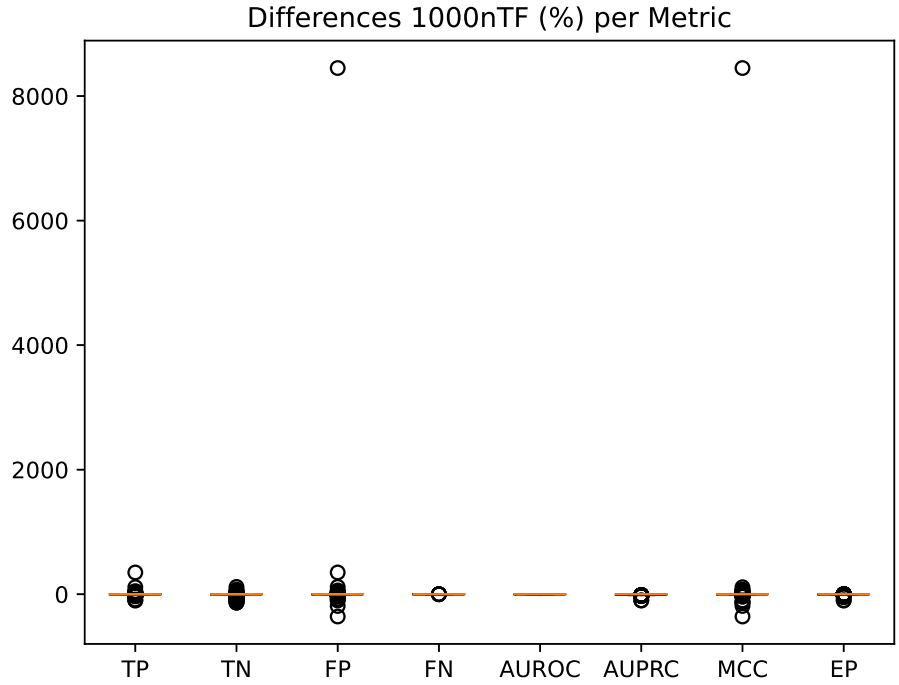
(a) Resultados para mHSC-E-1000nTF (b) Resultados para mHSC-L-1000nTF



Fonte: Elaborado pelo autor (2024).

quando um algoritmo de inferência tenta capturar a autorregulação, existe grande variação nos resultados. Posto isso e a importância desse *motif* de rede na manutenção de vida em diversos organismos, considerar a avaliação da autorregulação pode fornecer informação

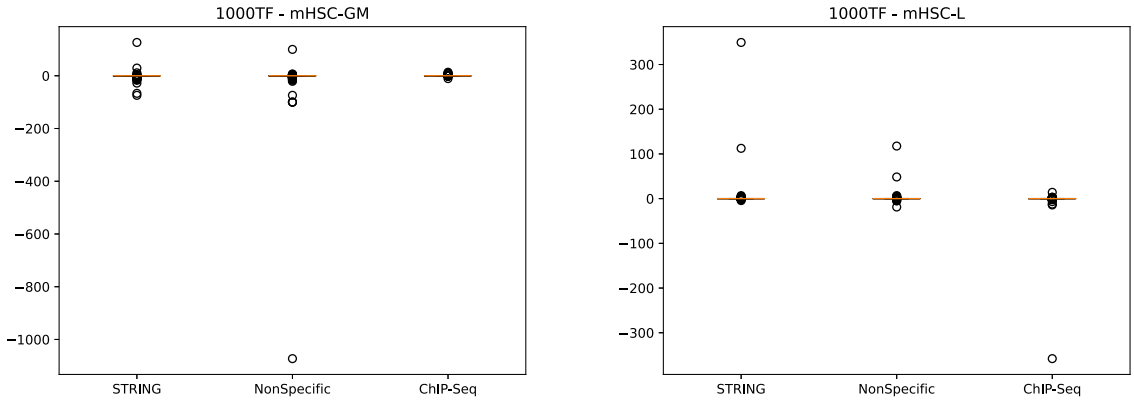
Figura 88 – Comparação de rede das diferenças entre as métricas para todos os problemas da configuração 1000nTF. A referência é sem autorregulação.



Fonte: Elaboardo pelo autor (2024).

Figura 89 – Resultados para os problemas considerando todas as métricas para as redes. A referência é sem autorregulação.

(a) Resultados para mHSC-GM-1000TF. (b) Resultados para mHSC-L-1000TF.



Fonte: Elaborado pelo autor (2024).

significante.

4.7.3 Avaliação de Desempenho

Tendo em vista o caso do MCC como métrica representativa para a análise de classificadores binários, iniciamos a análise comparando o MCC com o EP. Nos resultados tabulares apresentados no material suplementar, é possível encontrar diversos casos em que a avaliação do EP é inferior a outros algoritmos com menor número de TPs. Contudo, o mesmo não é observado para MCC. Alguns desses casos são resumidos a seguir:

- Para o problema hESC, sem autorregulação, na configuração 500nTF, para a rede *ChIP-Seq*, o algoritmo CGP-DSSPD-NR5-EXE1 apresenta 40TP e foi avaliado com 0,004866 em EP e 0,01064 em MCC. Contudo, SINCERITIES, para o mesmo caso, possui apenas 3TP e um EP de 0,005291. Contudo, MCC é 0,00319. Esse valor é inferior ao observado para o primeiro algoritmo. Uma situação similar é observada para o algoritmo CGP-DSSPD-NR10-EXE5. O mesmo acontece quando esses algoritmos são comparados considerando-se a autorregulação.

- Para o problema hHep, sem autorregulação, na configuração 500nTF, para a rede STRING, o algoritmo CGP-DSSPD-NR5-EXE1 apresenta 25TP e foi avaliado com 0,002755 em EP e 0,00544 em MCC. O algoritmo GRNBOOST2 apresenta 2 TP e um EP maior (0,004808) mas um menor MCC (0,00315).

- Para o problema mHSC-E, na configuração 500TF sem autorregulação, em relação à rede *ChIP-Seq*, o CGP-KM-BKM-NR10-EXE4 apresenta 137TP, 0,025277 EP, e 0,00136 MCC. O algoritmo PIDC, com 205 TP, apresenta menor EP (0,017737) e um maior MCC, em módulo (0,00571).

- Para o problema mHSC-E, na configuração 1000nTF, sem autorregulação, para a rede *NonSpecific*, o CGP-DSSPD-NR5-EXE4 alcançou 19 TP, 0,000984 EP e 0,00161 MCC. Já o algoritmo PIDC apresenta 17 TP, 0,024854 EP e 0,02419 MCC. Nesse caso, MCC alcança valores maiores no PIDC mesmo que com valores menores de TP e existe uma diferença grande (mais de 18.000) no número de TN.

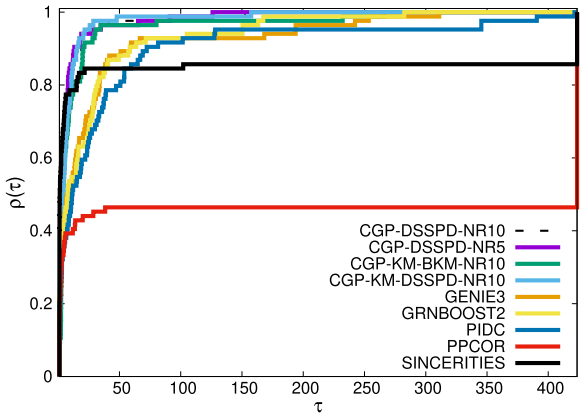
Uma possível explicação para essa variação nos valores de EP reside no fato de que mesmo que o EP considere apenas o TP, o *framework* BEELINE usa somente os genes apresentados na rede de referência ao invés de todo o conjunto de dados. Como resultado, todos os genes que foram incorretamente selecionados e não possuem relações regulatórias de acordo com a rede de referência, são ignorados no cálculo de EP.

Além disso, como previamente discutido, MCC leva em consideração as quatro medidas básicas da matriz de confusão. Como o problema de inferência de GRNs geralmente envolve dados desbalanceados, levar em consideração não só o TP, mas também TN, FP e FN, torna-se essencial para uma visão mais completa do desempenho dos algoritmos.

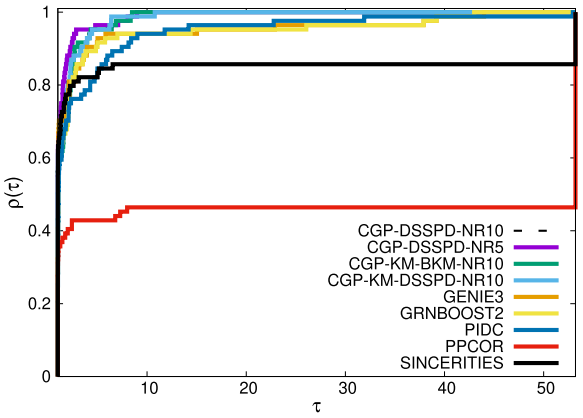
Também, podemos considerar o uso de PPs. A comparação entre as quatro métricas mais comuns da literatura, a saber AUROC, AUPRC, MCC e EP, são resumidas nos PPs apresentados nas Figuras 90 e 91 para os casos com e sem autorregulação, respectivamente. Resultados tabulares adicionais considerando todas as métricas e PPs para cada métrica individualmente estão disponíveis no material suplementar.

Figura 90 – Resultados para os problemas considerando todos os cenários sem autorregulação.

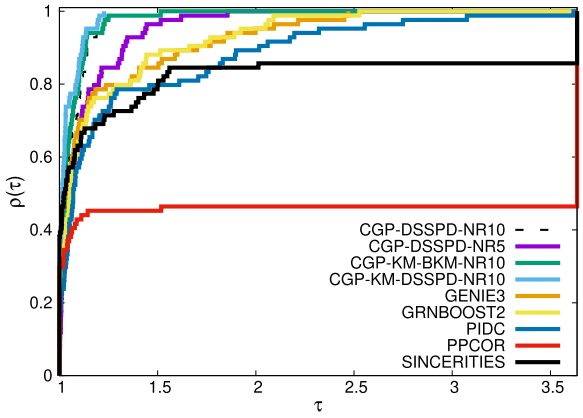
(a) Resultados considerando MCC



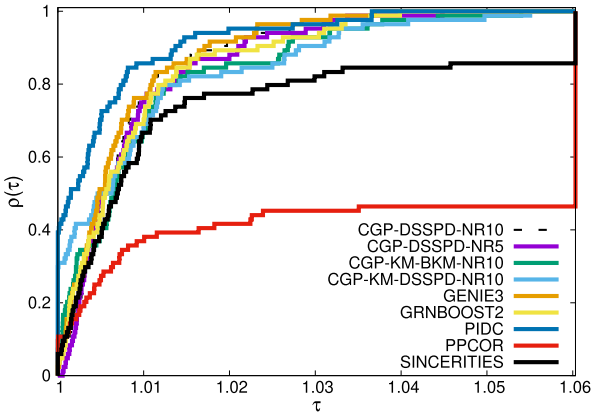
(b) Resultados considerando EP



(c) Resultados considerando AUPRC



(d) Resultados considerando AUROC



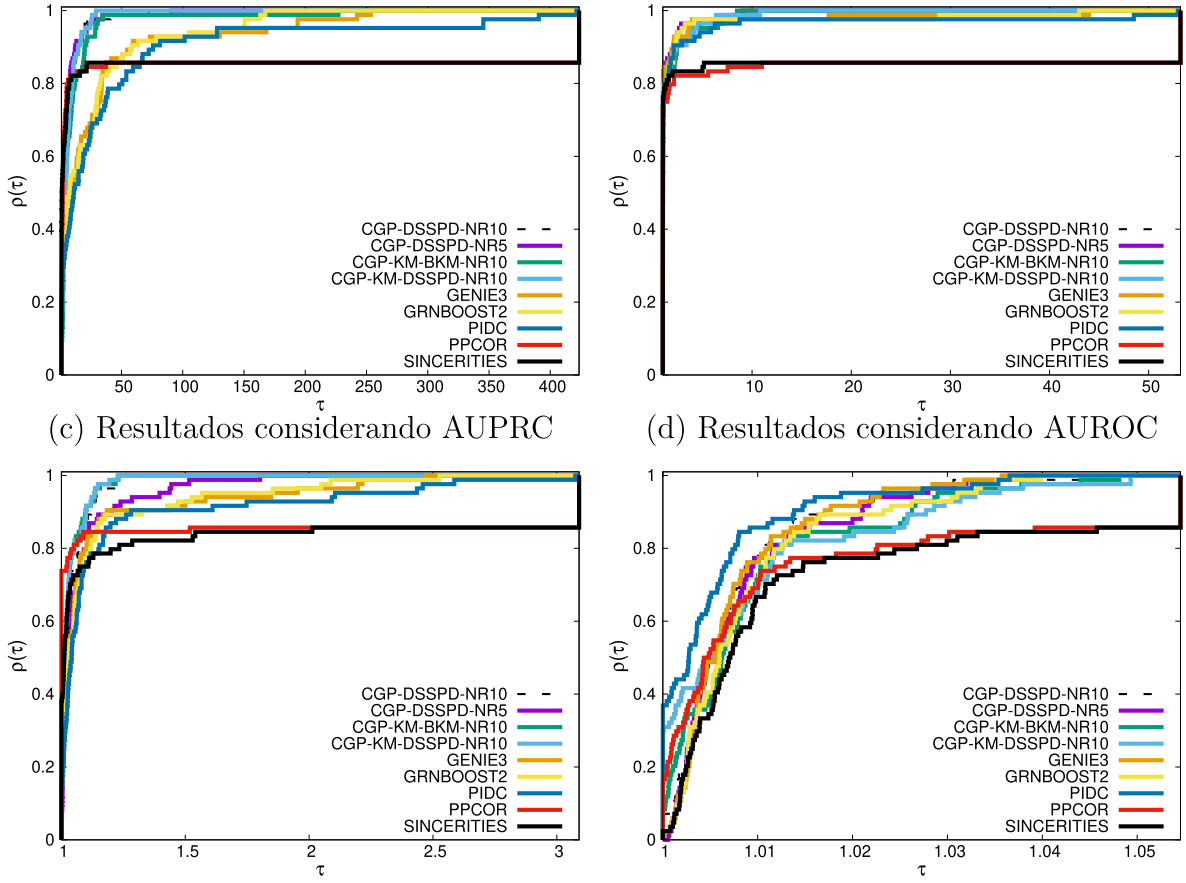
Fonte: Elaborado pelo autor (2024).

A Tabela 47 apresentam as áreas sob as curvas dos PPs normalizadas. Maiores áreas indicam melhor desempenho geral do algoritmo considerado. É possível perceber que os agoritmos que apresentam melhor desempenho geral não mudam quando a autorregulação é considerada ou não. Contudo, o melhor algoritmo é diferente a depender da métrica utilizada. Para o caso do MCC e EP, CGP-DSSPD-NR5 obteve o melhor resultado. Por outro lado, se considerada a AUPRC, o melhor algoritmo é CGP-KM-DSSPD-NR10. Por fim, se a métrica de referência é AUROC, o melhor algoritmo é o PIDC. Portanto, é claro que a métrica usada como referência leva a diferentes resultados no que diz respeito ao algoritmo com melhor desempenho. Devido ao desbalanceamento dos dados de expressão

Figura 91 – Resultados para todos os problemas considerando todos os cenários com autorregulação.

(a) Resultados considerando MCC

(b) Resultados considerando EP



(c) Resultados considerando AUPRC

(d) Resultados considerando AUROC

Fonte: Elaborado pelo autor (2024).

gênica, os problemas associados com a interpretação de AUPRC e AUROC, baseado nos estudos anteriores da literatura, bem como a defesa do MCC em relação as demais métricas, e o fato de que essa métrica leva em consideração as quatro medidas básicas da matriz de confusão, sugerimos aqui que o MCC seja utilizada como métrica padrão para a avaliação de GRNs.

É importante ressaltar que, para o algoritmo PPCOR, os resultados quando considerada a autorregulação são significativamente piores que aqueles obtidos quando a autorregulação é considerada. Contudo, essas diferenças não são significativas para os demais algoritmos.

4.8 DISCUSSÃO

Neste capítulo foram apresentados os experimentos computacionais relacionados ao desempenho do método proposto em diversos conjuntos de dados e os experimentos computacionais para análise e validação das etapas da proposta, além da avaliação do

Tabela 47 – Resultados para área sob a curva dos PPs para todas as métricas considerando ou não a autorregulação para todos os algoritmos.

Algoritmo	Com Autorregulação				Sem Autorregulação			
	MCC	EP	AUPRC	AUROC	MCC	EP	AUPRC	AUROC
CGP-DSSPD-NR10	0,9993	0,9974	0,9903	0,9366	0,9999	0,9973	0,9956	0,9448
CGP-DSSPD-NR5	**1,0000**	**1,0000**	0,9696	0,9263	**1,0000**	**1,0000**	0,9802	0,9332
CGP-KM-BKM-NR10	0,9858	0,9950	0,9937	0,9052	0,9903	0,9971	0,9996	0,9080
CGP-KM-DSSPD-NR10	0,9995	0,9962	**1,0000**	0,9102	0,9999	0,9976	**1,0000**	0,9131
GENIE3	0,9489	0,9650	0,9396	0,9526	0,9487	0,9890	0,9568	0,9462
GRNBOOST2	0,9571	0,9624	0,9419	0,9269	0,9563	0,9863	0,9580	0,9171
PIDC	0,9244	0,9607	0,8980	**1,0000**	0,9205	0,9758	0,9329	**1,0000**
PPCOR	0,4678	0,4632	0,4653	0,4478	0,8635	0,8572	0,8656	0,8368
SINCERITIES	0,8649	0,8600	0,8337	0,8039	0,8639	0,8607	0,8488	0,7960

Fonte: Elaborado pelo autor (2024).

processo metodológico da literatura quando considerado dados perfilados por scRNA-Seq.

Em relação ao desempenho do método proposto, no primeiro conjunto de experimentos, foram testados os problemas sintéticos, acurados e experimentais do benchmark de PRATAPA *et al.* (2020). Os resultados obtidos pelo CGPGRN para este cenário são superiores ou competitivos com os algoritmos estado da arte, considerando os testes estatísticos de Kruskal-Wallis (para mediana) e Dunn (para testes *post-hoc*). Os PPs mostram que, tanto no melhor caso quanto no caso da mediana, o CGPGRN obteve os melhores resultados para a maioria dos problemas, o melhor desempenho geral e é a abordagem mais confiável. Quando analisados os valores de BEELINE AUPRC para todos os problemas, a proposta apresentou melhores resultados em 33 de 42 problemas, sendo superior a todos os demais algoritmos. Já em relação ao BEELINE AUROC, CGPGRN obteve o segundo melhor resultado (29), ficando atrás de PIDC (31).

Já para os problemas experimentais, foi utilizada a métrica EP como referência, tendo em vista a impossibilidade de cálculo de AUPRC e AUROC pelo BEELINE por conta de uso excessivo de memória. É importante ressaltar que tal limitação foi resolvida no *framework* CGPGRN, além da inserção de outras métricas de avaliação para classificadores binários. Foram consideradas as três redes de referência (STRING, NonSpecific e ChIP-Seq) e as configurações 500nTF, 500TF, 1000nTF e 1000TF. Os resultados obtidos pela proposta na configuração 500nTF mostram o desempenho superior em 9/21 problemas, sendo o método com maior quantidade de melhores resultados, seguido pelo PIDC. Além disso, nenhum dos métodos foi capaz de encontrar relações regulatórias corretas em todos os problemas para todas as redes. Contudo, a proposta é a que apresenta menor incidência de tal situação (3). Já para a configuração 500TF, a proposta obteve a segunda maior quantidade de melhores resultados, ficando atrás do PIDC. Quando considerados os resultados da configuração 1000nTF, apesar do PIDC ter obtido melhores resultados em 6/21, a proposta obteve resultados melhores em 5/21 problemas. Por fim, para a

configuração 1000TF, PIDC obteve os melhores resultados em 13/21 problemas e a CGP, em segundo lugar, com 4/21 problemas. Em contagens absolutas, por PIDC apresenta os melhores resultados para a rede STRING e NonSpecific. Já em relação à rede ChIP-Seq, a proposta é melhor. Contudo, a proposta foi capaz de encontrar relações regulatórias corretas em mais problemas do que o PIDC. Uma possível explicação para o desempenho da proposta não ser o melhor em todos os casos reside no fato do método de avaliação empregado pelo BEELINE. Nele, somente as top-k relações regulatórias, sendo k a quantidade de relações regulatórias presentes na rede de referência, são consideradas. Além disso, somente os genes presentes nessa lista são considerados para o cálculo de TP, TN, FP e FN, o que não é apropriado, tendo em vista a presença de mais genes do que somente estes nos dados de expressão gênica de cada problema. Quando consideradas as redes completas, a proposta obtém melhores resultados em todos os casos. Além disso, é importante ressaltar que, dadas as características da tecnologia de perfilamento scRNA-Seq e a natureza da construção das redes ChIP-Seq, tal rede tende a fornecer maiores informações sobre os tipos celulares envolvidos do que as demais redes. A proposta é superior ao PIDC em diversos aspectos, principalmente naqueles relacionados à interpretabilidade do modelo. O CGPGRN é capaz de fornecer regras lógicas e sinal da regulação, enquanto PIDC fornece somente uma lista ordenada de possíveis reguladores.

Experimentos computacionais também foram realizados a fim de avaliar o desempenho da proposta em dados de organismos amplamente estudados, tais como a *S. cerevisiae* e a *E. Coli*, além de dados da competição DREAM4. Para a rede DNA SOS, a proposta obteve valores próximos a 0,99 em AUPRC e sempre superiores a 0,95 em AUROC. Tais resultados são sempre superiores aos obtidos pelo PIDC. Já para o problema IRMA, a proposta é sempre superior a todos os algoritmos estado da arte. Por fim, para os dados da competição DREAM4, em relação a AUPRC, a proposta apresenta os melhores resultados tanto para 10 quanto para 100 genes em todas as 5 redes. Já para AUROC, a proposta é melhor em todas as redes de 10 genes e resultados competitivos para a rede de 100 genes.

Por fim, a proposta foi analisada em relação a outros métodos Booleanos, especialmente em tempo computacional, incluíndo um que utiliza metaheurística (ATEN), constituindo uma abordagem mais próxima da proposta, onde os experimentos também levam em consideração o MCC. Para 16 e 32 genes, a proposta é melhor em 8/10 redes. Já para 64 genes, a proposta é melhor em 9/10 redes. Isso indica a superioridade da proposta em relação a métodos que modelam GRNs de forma Booleana através de metaheurísticas. Além disso, em relação ao tempo computacional, a proposta é a que apresenta a melhor escalabilidade dentre os métodos comparados. Também, por conta de tais problemas de escalabilidade associados à modelagem Booleana, todos os algoritmos limitam o número máximo de genes e a quantidade máxima de reguladores por gene. Tais restrições e limitações não existem na proposta.

A obtenção de um modelo contínuo a partir de um modelo Booleano, modelado

através de um sistema de equações diferenciais cujos coeficientes numéricos são determinados através de ES, também foi validada através de experimentos computacionais. A metodologia inicial foi corrigida e os dados foram divididos em conjuntos de treino e teste. Os resultados mostram que, diferentemente de uma técnica de GP tradicional, a proposta foi capaz de reproduzir corretamente o comportamento temporal tanto do modelo de ritmo circadiano de 5 espécies quanto o de 10 espécies.

Todos os experimentos anteriores objetivavam mostrar o comportamento e o desempenho da proposta frente aos algoritmos estado da arte para a inferência de GRNs. Contudo, a proposta é composta de diversas etapas. Cada uma dessas etapas afeta a obtenção de GRNs. Dessa forma, experimentos computacionais foram realizados para avaliar e melhorar as etapas do método proposto. Seguindo o fluxograma da proposta, inicialmente foram apresentados experimentos relacionados à importância do pré-processamento através do uso de *smoothing splines* e o efeito de tal etapa nos resultados da discretização. Os resultados indicam que o uso do pré-processamento proposto obteve melhores resultados em 43/60 situações para AUPRC e em 40/60 situações para AUROC. Isso reforça a importância de utilizar uma etapa de pré-processamento adequado e é um diferencial da proposta, tendo em vista que os métodos estado da arte já assumem que os dados estão prontos para uso.

Na sequência, foi analisada a importância do uso de uma etapa de agrupamento para direcionamento do processo de busca. Os resultados indicam que o uso do agrupamento via Kmeans, a evolução de um circuito único para todas as saídas e a discretização por *cluster* são capazes, em conjunto, de fornecer os melhores resultados, sendo a abordagem mais confiável e apresentando o melhor desempenho geral. Além disso, os resultados da proposta são sempre superiores aos obtidos pelos algoritmos estado da arte. Dentre as técnicas testadas, 7 abordagens que utilizam a CGP aparecem com desempenho geral superior antes do primeiro algoritmo estado da arte (PIDC), que ocupa a oitava posição. Tal etapa, apesar de opcional durante o processo da proposta, torna-se especialmente importante quando não existem informações sobre os melhores subconjuntos de genes a serem utilizados para a inferência.

O impacto de diversos operadores de discretização também foi avaliado. Inicialmente, utilizando como referência os operadores de discretização comumente utilizados na literatura para a discretização de GED, experimentos computacionais foram realizados para avaliar a qualidade das GRNs inferidas em cada cenário. Como alguns destes métodos possuem parâmetros (Top% e Max-X%Max), um estudo inicial para a determinação destes melhores parâmetros foi realizado. Os resultados mostram que o parâmetro de 50% maximiza os resultados obtidos por ambos os métodos de discretização. Tal parâmetro é utilizado para estes métodos de discretização e os demais métodos (Bikmeans, EFD, EWD, Gallo, Mean, Median, TSD e Kmeans, tanto no escopo de dados de linha quanto de matriz) foram comparados. Os resultados indicam que o EFD obteve o melhor desempenho

geral nos casos testados, seguido pela *Median*. Contudo, somente dados acurados foram utilizados para essa avaliação.

O método de discretização proposto (DSSPD) também foi analisado. Inicialmente, compara-se a CGP com Bikmeans e a CGP com a discretização proposta. Os resultados mostram que o DSSPD obteve os melhores resultados na maioria dos casos testados, tanto em AUPRC quanto em AUROC. Continuando as análises do DSSPD, a proposta é comparada com os algoritmos estado da arte GENIE3, GRNBOOST2, PIDC, PPCOR e SINCERITIES. Os resultados apresentados nos PPs indicam que, para o melhor caso, a proposta obteve os melhores resultados na maioria dos problemas e possui o melhor desempenho geral. Isso indica que a discretização proposta modela de maneira mais eficiente os estados discretos do que as demais abordagens de discretização.

Como muitos métodos de discretização estão disponíveis na literatura, experimentos preliminares foram conduzidos para avaliar a combinação de alguns destes em um método *ensemble*. Para tal, foram considerados o EFD, apontado anteriormente como o melhor método de discretização nos problemas acurados, o Bikmeans, amplamente utilizado na literatura e o DSSPD. Os resultados mostram que o *ensemble* não foi capaz de fornecer resultados melhores que os métodos de discretização comparados. Como esperado e apresentado anteriormente, o EFD obteve os melhores resultados para esses problemas acurados.

Entretanto, dados experimentais tendem a ser mais desafiadores para os métodos de discretização e inferência. Por esse motivo, experimentos computacionais foram realizados para determinar o desempenho do EFD, Bikmeans e DSSPD neste contexto. Os resultados mostram que o EFD, ainda que ressaltado anteriormente como o melhor para dados acurados apresentou os piores resultados nos dados experimentais. O DSSPD, além de sempre apresentar resultados melhores que os obtidos pelo EFD, é capaz de superar o Bikmeans em quase todos os problemas, especialmente com a rede de referência ChIP-Seq, e obtém resultados próximos, superando o Bikmeans em alguns problemas nas redes de referência STRING e NonSpecific.

Por fim, análises preliminares foram realizadas sobre o impacto de operadores de mutação na CGP para a obtenção de modelos Booleanos. O objetivo é descobrir se a obtenção de soluções factíveis de maneira mais rápida tem um impacto positivo na qualidade das soluções e a importância da otimização em termos de redução da complexidade do modelo através da minimização do número de elementos lógicos. Neste contexto, foram analisados os operadores SOMO, SAM e combinações destes dois. Como o SOMO é conhecido por ficar preso em mínimos locais, introduzimos uma abordagem com reinicialização da população a cada 1.000 avaliações da função objetivo quando não há melhoria nos valores de aptidão. Além disso, foram utilizadas a CGP com SAM, tradicionalmente utilizada nos experimentos deste trabalho, o SOMO tradicional, a

combinação de SOMO-SAM, onde SOMO é utilizado para obter a primeira solução factível e o SAM para otimização e essa mesma abordagem variando o parâmetro p_q, responsável por determinar a quantidade de nós inativos que sofrem mutação em suas entradas e funções. Inicialmente as análises são concentradas nestas abordagens que consideram a CGP. Os resultados mostram que as abordagens que utilizaram a etapa de otimização (SAM) obtiverem resultados melhores que as demais. O esquema de reinicialização auxiliou na exploração do espaço de busca e melhorou os resultados em relação à abordagem sem reinicialização. Contudo, não foi capaz de superar os resultados obtidos a partir da combinação SOMO-SAM. Além disso, o uso de $p_q = 50\%$ gerou resultados melhores que os apresentados na proposta original do SOMO. É possível perceber que a SOMO de fato auxilia na obtenção de soluções factíveis de maneira mais rápida, em termos de número de avaliações da função objetivo, e que o SAM é capaz de otimizar as soluções. Contudo, as soluções iniciais da SOMO possuem muitos elementos lógicos e isso torna-se um complicador para otimizar as soluções. Isso significa dizer que o SAM, quando aplicado tanto para obtenção de soluções factíveis quanto para otimização, gera modelos menos complexos, ainda que necessitando de um maior número de avaliações da função objetivo. As abordagens que obtiveram os melhores desempenhos nessa comparação são o SOMO-SAM e o SOMO-SAM-PQ50. Essas, por sua vez, são utilizadas na comparação com o GENIE3. Os resultados dessa comparação indicam que para AUPRC, GENIE3 apresenta os melhores resultados em todos os casos. Já para AUROC, SOMO-SAM é a abordagem mais confiável e apresenta o melhor desempenho geral. Os resultados inferiores obtidos pela proposta em relação ao GENIE3 podem estar relacionados ao uso do Bikmeans como método de discretização que, conforme apresentado anteriormente, pode não ser o método mais apropriado em todos os cenários. Esses experimentos preliminares indicam que modificar o operador de mutação da CGP pode auxiliar na obtenção de soluções factíveis com menor número de avaliações de função objetivo e que o SAM é apropriado para a redução da complexidade do modelo. Além disso, o SOMO é computacionalmente caro, pois requer a montagem de diversas tabelas verdade parciais para determinar o melhor nó que pode ser conectado ao nó que está sendo mutado. Esse fato pode não compensar a redução do número de avaliações da função objetivo para obter a primeira solução factível. Estudos adicionais ainda são necessários para determinar o melhor conjunto de operadores de mutação na CGP para a inferência de GRNs.

Já para a avaliação do processo metodológico em scRNA-Seq, considerando as práticas da literatura na seleção de genes, na modelagem de *motifs* de rede e a forma de avaliar o desempenho dos algoritmos de inferência e a qualidade das redes inferidas, diversos experimentos computacionais foram realizados a fim de verificar a eficiência dessas etapas.

Os experimentos iniciam-se na seleção de subconjuntos de genes. Primeiramente, utilizando os conjuntos de dados experimentais disponibilizados no *benchmark*, investigou-

se as quantidades de genes que compartilham relações regulatórias para cada uma das configurações (500nTF, 500TF, 1000nTF e 1000TF). Quando considerados os problemas na configuração 500nTF, percebe-se que, especialmente para a rede de referência STRING, o número de genes com relações regulatórias compartilhadas não é superior a 32,2%, no problema hHep. Em especial, o problema mDC, apresenta 6,8%, 6,6% e 17,4% de genes com relações regulatórias compartilhadas nas redes ChIP-Seq, NonSpecific e STRING, respectivamente. Além disso, a maior quantidade de relações regulatórias compartilhadas são apresentadas na rede ChIP-Seq. Já para a configuração 500TF, diferentes números de espécies são apresentados para cada problema, tendo em vista a consideração dos TFs. O número de espécies varia entre 560 e 1120. Novamente, a rede STRING é a que apresenta menor quantidade de relações regulatórias compartilhadas e a ChIP-Seq, a maior. Para a configuração 1000nTF, notou-se que o problema mHSC-L não cumpriu os requisitos do número de espécies, limitando-se a 692. Tanto as redes de referência NonSpecific quanto a STRING são as que apresentam menores quantidades de genes com relações regulatórias compartilhadas. A rede ChIP-Seq, novamente, apresenta a maior quantidade de relações regulatórias. Por fim, para a rede 1000TF, a mesma questão do número de espécies no problema mHSC-L é notado. A rede STRING é a que apresenta menor número de relações regulatórias compartilhadas, seguido pela NonSpecific e a STRING. De maneira geral, para todos os casos analisados, a rede ChIP-Seq é a rede capaz de apresentar a maior quantidade de relações regulatórias. Tendo em vista as características de que em muitos casos poucos genes compartilham relações regulatórias, foi avaliada a eficiência do uso de agrupamento, almejando maximizar esses genes que compartilham relações regulatórias. Para esse experimento, considerou-se a qualidade do agrupamento através das métricas homogeneidade, completude e *V-Measure*. Os resultados indicam que para todos os problemas e para todas as redes, ao menos um dos *clusters* aumentou o número de relações regulatórias. Além disso, a qualidade do agrupamento, em relação ao *V-Measure*, são sempre melhores na rede ChIP-Seq. Isso é um indicativo de que técnicas de agrupamento conseguem, de fato, concentrar grupos que compartilham relações regulatórias em diferentes *clusters*. O bom comportamento das técnicas de agrupamento e a alta quantidade de genes que compartilham relações regulatórias na rede ChIP-Seq é justificado pelo fato de que tal rede, em sua construção, leva em consideração o tipo celular envolvido no processo. Como a tecnologia de perfilamento scRNA-Seq fornece informações sobre células individuais, é possível que as redes ChIP-Seq sejam mais informativas nessa situação.

Uma segunda análise foi conduzida a fim de verificar a seleção de subconjuntos de genes de interesse, considerando-se os subconjuntos analisados pelos autores que disponibilizam os dados dos problemas hHep e hESC. Esses conjuntos foram intersectados com os *datasets* do *benchmark* e com as redes de referência. Para o subconjunto de 379 genes do hHep, somente 94% dos genes estavam presentes no *dataset* do *benchmark*. O mesmo

acontece para o problema hESC no subconjunto de 2.178 genes. O resultado da interseção dos subconjuntos apresentados nas publicações originais com as redes de referência mostram que, especialmente para o subconjunto de 2.036 espécies do problema hHep, o número de relações regulatórias compartilhadas é sempre superior à 77%, ficando acima dos 94% para as redes NonSpecific e STRING. Diferentemente do observado anteriormente, as redes NonSpecific e STRING apresentaram quantidades maiores de genes com relações regulatórias compartilhadas.

Comparamos também os subconjuntos de 355 e 150 genes, para os problemas hHep e hESC, com aqueles disponibilizados na configuração 500nTF. Ao considerar o problema hHep, a proporção de genes com relações regulatórias permanece similar, com variações em torno de 10%. Contudo, o número de relações regulatórias é menor para as redes NonSpecific e STRING. Já para o problema hESC, a proporção de genes com relações regulatórias é maior quando considera-se o conjunto de genes selecionados pelos experimentalistas. Em especial, na rede ChIP-Seq, 92% dos genes compartilham relações regulatórias segundo o subconjunto dos experimentalistas frente à 62% utilizando o GAM do *benchmark*. Aprofundando a análise na capacidade de seleção de subconjuntos de genes pelo GAM, geramos dados com o mesmo número de genes apresentados nos trabalhos originais (355 para hHep e 150, 2.036 e 3.247 para hESC). Ao intersectar os genes selecionados pelo GAM com os genes apresentados nas publicações originais, percebe-se que o GAM não foi capaz de encontrar mais do que 42% dos mesmos genes que os selecionados pelos experimentalistas. Isso se torna especialmente crítico no subconjunto de 150 genes para o problema hESC, onde somente 2% dos genes foram obtidos pelo GAM. Tais análises ressaltam que o critério utilizado pelo algoritmo de seleção pode diferir significativamente daqueles usados pelos experimentalistas, resultando em conjuntos diferentes. Além disso, a literatura ressalte que a variância, como utilizada no GAM, não pode ser usada como um indicador direto de HVGs.

Na sequência, foi analisada a consideração da autorregulação como *motif* de rede no momento de avaliação das GRNs inferidas. Para todos os casos analisados, percebe-se que as diferenças, na mediana, são em torno de zero, tanto para considerar ou não a autorregulação na avaliação. Contudo, os resultados indicam que *outliers* acontecem com frequência. Tal fato é justificado por conta do baixo número de TP. Uma GRN que anteriormente possuía um único TP e numa segunda avaliação passa para dois TP gera uma diferença de 100%. De maneira análoga, esse fato é observado para as demais métricas comparadas, como o MCC. Como a literatura ressalta a importância da autorregulação para a manutenção de vida e que os resultados não são afetados significativamente entre considerar ou não tal *motif*, recomenda-se considerar a autorregulação na avaliação.

Por fim, estudos foram conduzidos a fim de determinar o melhor conjunto de métricas a serem considerados para a avaliação de classificadores binários, especialmente para dados desbalanceados, como no caso de GRNs. O conjunto de métricas apresentado

pela literatura nesse contexto é bastante variado. Contudo, no contexto de classificadores binários, o MCC considera todas as métricas básicas da matriz de confusão. Tendo isso em vista, os experimentos computacionais foram realizados comparando-se as métricas comumente utilizadas na literatura, tais como AUPRC e AUROC, o EP, e o MCC. O EP considera somente a fração de verdadeiros positivos obtidos. Contudo, os experimentos mostram que o EP pode fornecer informações discordantes em relação ao número de TPs. Por exemplo, alguns casos onde um algoritmo gera uma rede com 40 TP apresenta EP inferior à outra rede obtida por outro algoritmo com somente 3 TP. Isso é observado para todas as configurações (500nTF, 500TF, 1000nTF, 1000TF), em todas as redes de referência, para a maioria dos problemas. Contudo, o MCC não apresenta tal discordância. Uma possível explicação para essas informações discordantes geradas pelo EP reside no fato de que somente os genes apresentados na rede de referência são utilizados no BEELINE, e não todos os genes apresentados no conjunto de dados. Isso pode não ser adequado. Dessa forma, sugere-se a adoção do MCC como métrica padrão para a avaliação de GRNs. Além disso, em avaliações onde consideram-se diversos problemas para diversas configurações, como na maioria dos casos apresentados nesta tese, o uso de uma única métrica e uma contagem absoluta da quantidade de vezes que um algoritmo gera a maior quantidade de maiores valores para essa métrica, pode não ser informativo o suficiente. Por esse motivo, sugerimos também a adoção dos PPs, tendo em vista que podem fornecer informações sobre o algoritmo com melhor resultado para a maioria dos problemas, o algoritmo mais confiável e o algoritmo com maior desempenho geral. Os PPs foram utilizados para comparar abordagens de CGP e os algoritmos estado da arte sob a visão das métricas MCC, EP, AUPRC e AUROC. Os resultados indicam que, a depender da métrica considerada, o melhor algoritmo varia. Por exemplo, para MCC e EP, a proposta, utilizando a discretização proposta é melhor. Já para AUPRC, os resultados são melhores quando utiliza-se a proposta, com a discretização proposta e a etapa de agrupamento. Por outro lado, quando considerada a AUROC, PIDC apresenta o melhor desempenho geral.

Ainda que o objetivo dessa revisão do processo metodológico seja analisar o que a literatura faz e fornecer sugestões para as etapas discutidas, os resultados apresentados em relação à proposta são, cronologicamente, os últimos realizados. Dessa forma, é importante ressaltar que, tanto para o EP, considerada métrica padrão para a avaliação de dados experimentais no BEELINE, quanto para o MCC, o *framework* proposto utilizando a discretização proposta obteve os melhores resultados em todos os cenários considerados nas análises. Além disso, destaca-se que não foi necessário o uso da etapa de agrupamento para a obtenção desses melhores resultados. Uma possível justificativa para isso é que o próprio DSSPD, ao levar em consideração a distribuição dos dados de expressão gênica, já realize, indiretamente, esse agrupamento entre dados com expressão gênica significativa e aqueles que estão sempre próximos de zero.

5 CONCLUSÕES E TRABALHOS FUTUROS

Modelos de biologia sistêmica podem ser utilizados para testar novas hipóteses formuladas utilizando conhecimento prévio ou dados resultantes de experimentação. Um dos principais problemas em biologia sistêmica é a inferência de redes de regulação gênica (GRNs), consistindo de uma tarefa difícil e um desafio científico amplamente abordado. Tais modelos podem ser utilizados para descrever e predizer dependências entre entidades moleculares. Neste contexto, o sequenciamento de RNas de célula única (scRNA-Seq) forneceu uma resolução sem precedentes para o campo da transcriptômica pois as células individuais podem ser analisadas de forma abrangente e imparcial. Os experimentos usando scRNA-Seq são atrativos para a inferência de GRNs devido à produção de milhares de medidas independentes e a possibilidade de organizar as células ao longo de trajetórias que descrevem o desenvolvimento ou progresso da célula através de algoritmos de inferência.

Existe um interesse crescente na aplicação de abordagens Booleanas em biologia sistêmica pois são menos suscetíveis a ruídos. Além disso, avanços na estimação de GRNs levaram a um maior entendimento da regulação celular. Contudo, avanços metodológicos adicionais ainda são necessários.

Neste trabalho é proposto um procedimento para a inferência de modelos de GRNs a partir de dados de expressão gênica perfilados por scRNA-Seq na forma de séries temporais onde: (i) os dados são pré-processados e discretizados, (ii) um modelo Booleano é criado a partir de Programação Genética Cartesiana, (iii) um modelo contínuo na forma de um sistema de equações diferenciais ordinárias é obtido a partir do modelo booleano, e (iv) os coeficientes numéricos do sistema de equações diferenciais são otimizados utilizando Estratégias Evolutivas. Além disso, um novo método de discretização para dados de expressão gênica, uma etapa de agrupamento para direcionamento do processo de busca e uma proposta adicional sobre o processo metodológico de inferência e avaliação de GRNs são apresentados.

Técnicas de computação evolucionista são conhecidas por apresentar problemas de escalabilidade, principalmente no que concerne a avaliação dos indivíduos. Por esse motivo, o uso de paralelismo e técnicas de computação de alto desempenho podem auxiliar na redução de tempo computacional para a obtenção de modelos de GRNs. Tendo isso em mente, a proposta contempla uma adaptação de uma implementação de CGP originalmente desenvolvida para a evolução de redes neurais artificiais, para inferir modelos de GRNs. Essa abordagem permitiu a exploração de dados experimentais, o que era inviável anteriormente devido à grande quantidade de genes.

O método proposto foi avaliado em dados simulados do ritmo circadiano da *Drosophila*, com a obtenção do modelo contínuo e os experimentos computacionais indicam que o método proposto é capaz de gerar corretamente e reproduzir a dinâmica das relações

regulatórias das GRNs, diferentemente do que é obtido ao utilizar técnicas tradicionais de Programação Genética.

Além disso, ao utilizar o procedimento proposto até a etapa de geração do modelo Booleano, que constitui um produto de grande interesse prático e da literatura, foi avaliado em problemas *benchmark*, organismos amplamente estudados na literatura e dados de competição. Os resultados foram comparados com as redes *ground-truth* ou redes de referência, para o caso dos problemas experimentais, e avaliados segundo as métricas de precisão inicial, área sob as curvas ROC e *precision-recall* e MCC, comumente adotadas na literatura. Os perfis de desempenho avaliados sob esses resultados mostram que o método proposto obteve os melhores resultados para a maioria dos problemas e obteve o melhor desempenho geral. Ainda, o método proposto é a abordagem mais confiável (obteve os melhores resultados no pior caso). Estes conjuntos de experimentos reforçam que a estratégia proposta consegue superar os algoritmos estado da arte. Além disso, em relação ao tempo computacional de modelos Booleanos, os resultados indicam que o método proposto é o que apresenta não só o menor tempo para a inferência das redes para todos os casos, mas também é o que apresenta a melhor escalabilidade, apresentando comportamento linear. Ainda, os algoritmos de inferência Booleana apresentados são capazes de lidar com no máximo 150 genes e com uma quantidade máxima de reguladores de 60 genes. Tais limites e restrições não existem no CGPGRN.

Dessa forma, ainda que a literatura não coloque modelos Booleanos dentre os algoritmos estado da arte, o CGPGRN constitui uma alternativa superior, em muitos casos, aos demais algoritmos. Isso ressalta, ainda, a superioridade do CGPGRN dentre os métodos que modelam GRNs na forma Boolena.

A importância das etapas da proposta também foram investigadas, onde é possível concluir que: (i) em relação aos métodos de discretização da literatura, em problemas sintéticos e acurados, EFD provê o melhor desempenho geral, tanto para AUPRC quanto AUROC, já para problemas experimentais o mesmo não é observado, e o DSSPD fornece os melhores resultados, (ii) para o pré-processamento, o uso de suavização via *smoothing splines* fornece melhores resultados, (iii) o agrupamento auxilia na obtenção de GRNs corretas, principalmente quando não existe conhecimento sobre o subconjunto de genes que devem ser utilizados para modelar o fenômeno biológico em questão, (iv) a modificação do operador de mutação da CGP para um SOMO com p_q 50% auxilia a CGP a obter melhores resultados, uma vez que uma descendência mais diversa é gerada.

Contudo, faz-se necessária uma investigação mais profunda sobre as relações entre as características dos dados e os métodos de discretização. Isso foi observado, por exemplo, para dados cíclicos, onde o TSD tende a apresentar melhores resultados. Também, explorar discretizações com mais de dois níveis de discretização podem auxiliar na determinação de estados intermediários e, consequentemente, refinar as informações extraídas dos dados de

expressão gênica.

Em relação ao agrupamento, outros métodos além do *k-means*, tais como os baseados em densidade, ainda precisam ser mais explorados.

Já para o operador de mutação da CGP, apesar da combinação SOMO e SAM melhorar a qualidade das GRNs inferidas quando comparada à estratégia do SOMO original, os dois conjuntos de experimentos iniciais, tanto com ou sem a etapa de obtenção do modelo contínuo, somente com o SAM, mostrou-se superior aos algoritmos estado da arte. Existe ainda a necessidade de maior exploração dos operadores de movimento da CGP no contexto de inferência de GRNs.

Há a necessidade de estudos futuros e desenvolvimento de técnicas apropriadas para pré-processamento, inferência mais confiável de *pseudotimes* ou outras tecnologias que possam fornecer informação temporal tal como o *RNA-Velocity* uma vez que tal informação tende a melhorar a qualidade dos dados e pode auxiliar na inferência de redes mais acuradas. Além disso, a integração de dados multi-ômicos apresentam um caminho promissor para a inferência de GRNs com maior interpretabilidade e significado para experimentalistas.

Em relação à avaliação do processo metodológico em scRNA-Seq, é possível concluir que: (i) o uso de medidas de variância para a identificação de HVGs fornece conjuntos de dados compostos por genes que nunca compartilham relações regulatórias, indicando que o GAM não é uma boa escolha para tal tarefa, (ii) não existe diferença na avaliação da qualidade da GRN inferida ao considerar, ou não, a autorregulação (iii) a depender da métrica utilizada como referência para avaliação da GRN, o algoritmo com melhor desempenho varia substancialmente, (iv) o uso da rede ChIP-Seq como rede de referência tende a fornecer maiores informações, uma vez que considera as relações regulatórias dependentes do tipo celular considerado, (v) o uso de problemas sintéticos e acurados servem como fonte de informação inicial sobre o desempenho de algoritmos de inferência mas os resultados não podem ser extrapolados para *datasets* de problemas reais

Baseado nessas conclusões, para a seleção de genes sugere-se o uso de técnicas de agrupamento quando não existe conhecimento biológico *a priori* dos genes que modelam o fenômeno biológico em questão. Em relação ao *motif* de autorregulação, sugere-se: (i) se sabe-se que o fenômeno biológico que está sendo modelado não apresenta relações regulatórias de autorregulação, tal *motif* de rede não precisa ser considerado, e (ii) usar a autorregulação sempre que não houver conhecimento sobre a existência de tal *motif* de rede no fenômeno biológico modelado. Já para as métricas de avaliação, tendo em vista que o MCC contempla as principais medidas da matriz de confusão, sugere-se a adoção de tal métrica como padrão para avaliação de GRNs. Sugere-se, também, o uso de rede de ChIP-Seq como rede de referência para experimentos que utilizam dados scRNA-Seq. Ainda, como tais redes de referência são muitas vezes construídas a partir

de experimentação biológica, e tendo em vista que algumas interações gênicas já são conhecidas pela literatura, tal informação pode ser utilizada como restrição do algoritmo de inferência a fim de direcionar o processo de busca.

É importante ressaltar que diversos outros *motifs* de rede estão presentes em diversos organismos e são relatados na literatura, tais como ciclos de realimentação. Para o melhor de nosso conhecimento, nenhum algoritmo modela tais *motifs* e esforços nesse sentido são necessários. Além disso, há a necessidade de estudo e desenvolvimento de novos métodos que sejam capazes de gerar dados sintéticos que efetivamente reproduzam as características dos dados experimentais de scRNA-Seq.

Apesar de nossas recomendações, esforços adicionais ainda são necessários para o desenvolvimento de métodos para a inferência de GRNs.

O *framework* CGPGRN proposto e apresentado nesta tese vai na direção de tentar resolver os problemas levantados na revisão do processo metodológico apresentado, na adoção de uma etapa de pré-processamento adequado, no uso de técnicas de agrupamento para a seleção de subconjuntos de genes e na seleção apropriada de uma rede de referência e métrica de avaliação. Com isso, espera-se que o *framework* CGPGRN auxilie no avanço e desenvolvimento de algoritmos de inferência de GRNs e que a revisão do processo metodológico apresentado auxilie pesquisadores que lidam com essa classe de problemas.

Além disso, todos os códigos desenvolvidos estão disponíveis publicamente, contendo documentação, resolução de problemas tanto na instalação quanto no uso, bem como *interface* gráfica e determinação automática dos parâmetros, a fim de facilitar a universalização do uso do *framework* CGPGRN.

Como trabalhos futuros, destacam-se: (i) integração de múltiplos dados ômicos, capazes de enriquecer os modelos, (ii) introdução de ciclos de retroalimentação, tanto positiva quanto negativa, pois a literatura afirma que são comuns em todas as células, (iii) adoção paralelismo em todo o processo do método proposto (desde o pré-processamento até a obtenção do modelo contínuo), (iv) avaliação de métodos de discretização com mais de 2 níveis de discretização, (v) exploração de métodos *ensemble* para a discretização de dados de expressão gênica, como os que utilizam meta aprendizado, (vi) exploração das características dos dados e sua relação com o desempenho dos métodos de discretização, (vii) desenvolvimento de um sistema de introdução de restrições, onde relações regulatórias conhecidas da literatura podem ser introduzidas *a priori* para auxiliar o algoritmo de inferência, através de gramáticas formais, (viii) explorar o conhecimento biológico que pode ser extraído a partir do modelo obtido em forma simbólica (ix) exploração dos parâmetros da SOMO, considerando outros valores de p_q e p_f no contexto de GRNs, (x) utilização de diagramas de decisão binária e redução por SAT como mecanismo de avaliação dos indivíduos da CGP, pois essa abordagem se mostrou extremamente eficiente no projeto evolutivo de circuitos digitais e, se possível, realizar isso utilizando o mesmo paralelismo

em dois níveis da abordagem paralela adotada neste trabalho, e (xi) exploração das redes geradas e avaliação das características da rede, tais como nós fortemente conectados.

REFERÊNCIAS

AALTO, A. *et al.* Gene regulatory network inference from sparsely sampled noisy data. **Nature Communications**, Nature Publishing Group, v. 11, n. 1, p. 1–9, 2020.

ADEREM, A. Systems biology: its practice and challenges. **Cell**, Elsevier, v. 121, n. 4, p. 511–513, 2005.

AIBAR, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. **Nature Methods**, Nature Publishing Group, v. 14, n. 11, p. 1083–1086, 2017.

AKERS, K.; MURALI, T. Gene regulatory network inference in single-cell biology. **Current Opinion in Systems Biology**, Elsevier, v. 26, p. 87–97, 2021.

AKOGLU, H. User's guide to correlation coefficients. **Turkish journal of emergency medicine**, Elsevier, v. 18, n. 3, p. 91–93, 2018.

ALANNI, R. *et al.* Deep gene selection method to select genes from microarray datasets for cancer classification. **BMC bioinformatics**, BioMed Central, v. 20, n. 1, p. 1–15, 2019.

ALBERTS, B. *et al.* **Biologia molecular da célula**. [S.l.]: Artmed Editora, 2010.

ALM, E.; ARKIN, A. P. Biological networks. **Current opinion in structural biology**, Elsevier, v. 13, n. 2, p. 193–202, 2003.

ALMEIDA, J. S.; VOIT, E. O. Neural-network-based parameter estimation in s-system models of biological networks. **Genome Informatics**, Japanese Society for Bioinformatics, v. 14, p. 114–123, 2003.

ANDO, S.; IBA, H. Construction of genetic network using evolutionary algorithm and combined fitness function. **Genome Informatics**, Japanese Society for Bioinformatics, v. 14, p. 94–103, 2003.

ANDRADE, T. P. d. **Interações gênicas usando redes booleanas limiarizadas modeladas como um problema de satisfação de restrições**. Tese (Doutorado) — Universidade de São Paulo, 2012.

ANDREWS, T. S. *et al.* Tutorial: guidelines for the computational analysis of single-cell rna sequencing data. **Nature Protocols**, Nature Publishing Group, p. 1–9, 2020.

ARABIE, P.; HUBERT, L.; SOETE, G. D. **Clustering and classification**. [S.l.]: World Scientific, 1996.

AUBIN-FRANKOWSKI, P.-C.; VERT, J.-P. Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference. **Bioinformatics**, Oxford University Press, v. 36, n. 18, p. 4774–4780, 2020.

AUGUSTO, D. A.; BARBOSA, H. J. Symbolic regression via genetic programming. In: IEEE. **Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks**. [S.l.], 2000. p. 173–178.

AUGUSTO, D. A.; BARBOSA, H. J. Accelerated parallel genetic programming tree evaluation with opencl. **Journal of Parallel and Distributed Computing**, Elsevier, v. 73, n. 1, p. 86–100, 2013.

BACK, T. **Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms**. [S.l.]: Oxford university press, 1996.

BÄCK, T.; FOUSSETTE, C.; KRAUSE, P. **Contemporary evolution strategies**. [S.l.]: Springer, 2013.

BARBOSA, H. J. C.; BERNARDINO, H. S.; BARRETO, A. M. S. Using performance profiles to analyze the results of the 2006 CEC constrained optimization competition. In: IEEE. **Evolutionary Computation (CEC), 2010 IEEE Congress on**. [S.l.], 2010. p. 1–8.

BARMAN, S.; KWON, Y.-K. A novel mutual information-based boolean network inference method from time-series gene expression data. **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 2, p. e0171097, 2017.

BARMAN, S.; KWON, Y.-K. A boolean network inference from time-series gene expression data using a genetic algorithm. **Bioinformatics**, Oxford University Press, v. 34, n. 17, p. i927–i933, 2018.

BATTITI, R. Using mutual information for selecting features in supervised neural net learning. **IEEE Transactions on neural networks**, IEEE, v. 5, n. 4, p. 537–550, 1994.

BECQUET, C. *et al.* Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data. **Genome Biology**, BioMed Central, v. 3, n. 12, p. research0067–1, 2002.

BERGER, V. W.; ZHOU, Y. Kolmogorov–smirnov test: Overview. **Wiley statsref: Statistics reference online**, Wiley Online Library, 2014.

BERNARDINO, H. S.; BARBOSA, H. J. C. Comparing two ways of inferring a differential equation model via grammar-based immune programming. In: **Proc. of the Iberian-Latin-American Congress on Computational Methods in Engineering**. [S.l.: s.n.], 2010.

BERNARDINO, H. S.; BARBOSA, H. J. C. Inferring systems of ordinary differential equations via grammar-based immune programming. In: SPRINGER. **Int. Conf. on Artificial Immune Systems**. [S.l.], 2011. p. 198–211.

BERTSIMAS, D.; TSITSIKLIS, J. Simulated annealing. **Statistical science**, Institute of Mathematical Statistics, v. 8, n. 1, p. 10–15, 1993.

BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies–a comprehensive introduction. **Natural computing**, Springer, v. 1, n. 1, p. 3–52, 2002.

BOLÓN-CANEDO, V.; SÁNCHEZ-MAROÑO, N.; ALONSO-BETANZOS, A. A review of feature selection methods on synthetic data. **Knowledge and information systems**, Springer, v. 34, p. 483–519, 2013.

BOLÓN-CANEDO, V. *et al.* A review of microarray datasets and applied feature selection methods. **Information sciences**, Elsevier, v. 282, p. 111–135, 2014.

BOMMERT, A. *et al.* Benchmark for filter methods for feature selection in high-dimensional classification data. **Computational Statistics & Data Analysis**, Elsevier, v. 143, p. 106839, 2020.

BRACCINI, M.; MONTAGNA, S.; ROLI, A. Self-loops favour diversification and asymmetric transitions between attractors in boolean network models. In: SPRINGER. **Artificial Life and Evolutionary Computation: 13th Italian Workshop, WIVACE 2018, Parma, Italy, September 10–12, 2018, Revised Selected Papers 13**. [S.l.], 2019. p. 30–41.

BRAYTON, R. K. *et al.* **Logic minimization algorithms for VLSI synthesis**. [S.l.]: Springer Science & Business Media, 1984.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

BRUICE, P. Y. **Química Orgânica - Vol. 1 e 2**. [S.l.]: Pearson Prentice Hall, São Paulo, 2006.

BUETTNER, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. **Nat. Biotechnol.**, Nature Publishing Group, v. 33, n. 2, p. 155–160, 2015.

BULCKE, T. Van den *et al.* Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. **BMC bioinformatics**, Springer, v. 7, p. 1–12, 2006.

BYRON, K.; WANG, J. T. A comparative review of recent bioinformatics tools for inferring gene regulatory networks using time-series expression data. **International journal of data mining and bioinformatics**, Inderscience Publishers (IEL), v. 20, n. 4, p. 320–340, 2018.

CAMBRON, M. *et al.* White-matter astrocytes, axonal energy metabolism, and axonal degeneration in multiple sclerosis. **Journal of Cerebral Blood Flow & Metabolism**, SAGE Publications Sage UK: London, England, v. 32, n. 3, p. 413–424, 2012.

CAMP, J. G. *et al.* Multilineage communication regulates human liver bud development from pluripotency. **Nature**, Nature Publishing Group UK London, v. 546, n. 7659, p. 533–538, 2017.

CANBEK, G. *et al.* Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In: IEEE. **2017 International Conference on Computer Science and Engineering (UBMK)**. [S.l.], 2017. p. 821–826.

CANTONE, I. *et al.* A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. **Cell**, Elsevier, v. 137, n. 1, p. 172–181, 2009.

CHAN, T. E.; STUMPF, M. P.; BABTIE, A. C. Gene regulatory network inference from single-cell data using multivariate information measures. **Cell Systems**, Elsevier, v. 5, n. 3, p. 251–267, 2017.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 1, p. 16–28, 2014.

CHARGAFF, E.; LIPSHITZ, R.; GREEN, C. Composition of the desoxypentose nucleic acids of four genera of sea-urchin. **Journal of Biological Chemistry**, Elsevier, v. 195, n. 1, p. 155–160, 1952.

CHEN, G.; NING, B.; SHI, T. Single-cell rna-seq technologies and related computational data analysis. **Frontiers in genetics**, Frontiers, v. 10, p. 317, 2019.

CHEN, S.; MAR, J. C. Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. **BMC Bioinformatics**, BioMed Central, v. 19, n. 1, p. 1–21, 2018.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. **BMC genomics**, BioMed Central, v. 21, n. 1, p. 1–13, 2020.

CHICCO, D.; JURMAN, G. The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification. **BioData Mining**, BioMed Central, v. 16, n. 1, p. 1–23, 2023.

CHICCO, D.; TÖTSCH, N.; JURMAN, G. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. **BioData mining**, BioMed Central, v. 14, n. 1, p. 1–22, 2021.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. **IEEE Access**, IEEE, v. 9, p. 78368–78381, 2021.

CHLEBUS, B. S.; NGUYEN, S. H. On finding optimal discretizations for two attributes. In: SPRINGER. **International Conference on Rough Sets and Current Trends in Computing**. [S.l.], 1998. p. 537–544.

CHU, L.-F. *et al.* Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. **Genome biology**, Springer, v. 17, p. 1–20, 2016.

CIOS, K. J.; PEDRYCZ, W.; SWINIARSKI, R. W. Data mining and knowledge discovery. In: **Data mining methods for knowledge discovery**. [S.l.]: Springer, 1998. p. 1–26.

CLARK, D. P.; PAZDERNIK, N. J. **Molecular biology**. [S.l.]: Elsevier, 2013.

CLEGG, J.; WALKER, J. A.; MILLER, J. F. A new crossover technique for cartesian genetic programming. In: ACM. **Proc of the Conf. on Genetic and Evolutionary Computation**. [S.l.], 2007. p. 1580–1587.

COHEN, I. *et al.* **Noise reduction in speech processing**. [S.l.]: Springer, 2009.

CORONA, I. *et al.* Information fusion for computer security: State of the art and open issues. **Information Fusion**, Elsevier, v. 10, n. 4, p. 274–284, 2009.

CRICK, F. H. On protein synthesis. In: **Symp Soc Exp Biol**. [S.l.: s.n.], 1958. v. 12, n. 138-63, p. 8.

DARWIN, C. **On the Origin of Species by Means of Natural Selection, 1859**. [S.l.]: Culture et Civilisation, 1969.

DAVIDSON, E. H. *et al.* A genomic regulatory network for development. **science**, American Association for the Advancement of Science, v. 295, n. 5560, p. 1669–1678, 2002.

DEB, K. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, v. 186, p. 311–338, 2000.

DELGADO, F. M.; GÓMEZ-VELA, F. Computational methods for gene regulatory networks reconstruction and analysis: a review. **Artificial intelligence in medicine**, Elsevier, v. 95, p. 133–145, 2019.

DENG, H.; RUNGER, G. Feature selection via regularized trees. In: IEEE. **The 2012 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2012. p. 1–8.

DESHPANDE, A. *et al.* Network inference with granger causality ensembles on single-cell transcriptomic data. **BioRxiv**, Cold Spring Harbor Laboratory, p. 534834, 2019.

DIETTERICH, T. G. Ensemble methods in machine learning. In: SPRINGER. **International workshop on multiple classifier systems**. [S.l.], 2000. p. 1–15.

DIMITROVA, E. S. *et al.* Discretization of time series data. **Journal of Computational Biology**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 17, n. 6, p. 853–868, 2010.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Math. Program.**, v. 91, n. 2, p. 201–213, Jan 2002.

DOLNICAR, S. A review of unquestioned standards in using cluster analysis for data-driven market segmentation. 2002.

DOUGHERTY, J.; KOHAVI, R.; SAHAMI, M. Supervised and unsupervised discretization of continuous features. In: **Machine learning proceedings 1995**. [S.l.]: Elsevier, 1995. p. 194–202.

EIBEN, A. E.; SMITH, J. E. *et al.* **Introduction to Evolutionary Computing**. [S.l.]: Springer, 2003.

ELSON, D.; CHARGAFF, E. On the desoxyribonucleic acid content of sea urchin gametes. **Experientia**, Springer, v. 8, n. 4, p. 143–145, 1952.

ERDAL, S. *et al.* A time series analysis of microarray data. In: IEEE. **Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering**. [S.l.], 2004. p. 366–375.

FEREA, T. L. *et al.* Systematic changes in gene expression patterns following adaptive evolution in yeast. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 96, n. 17, p. 9721–9726, 1999.

FINAK, G. *et al.* Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. **Genome biology**, BioMed Central, v. 16, n. 1, p. 1–13, 2015.

FRALEY, C.; RAFTERY, A. E. How many clusters? which clustering method? answers via model-based cluster analysis. **The computer journal**, Oxford University Press, v. 41, n. 8, p. 578–588, 1998.

GALLO, C. A.; CARBALLIDO, J. A.; PONZONI, I. Discovering time-lagged rules from microarray data using gene profile classifiers. **BMC bioinformatics**, Springer, v. 12, n. 1, p. 1–21, 2011.

GALLO, C. A. *et al.* Discretization of gene expression data revised. **Briefings in Bioinformatics**, Oxford University Press, v. 17, n. 5, p. 758–770, 2015.

GAMA-CASTRO, S. *et al.* Regulondb version 7.0: transcriptional regulation of escherichia coli k-12 integrated within genetic sensory response units (gensor units). **Nucleic acids research**, Oxford University Press, v. 39, n. suppl_1, p. D98–D105, 2010.

GAO, N. P. *et al.* Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. **Bioinformatics**, Oxford University Press, v. 34, n. 2, p. 258–266, 2018.

GARCIA-ALONSO, L. *et al.* Benchmark and integration of resources for the estimation of human transcription factor activities. **Genome research**, Cold Spring Harbor Lab, v. 29, n. 8, p. 1363–1375, 2019.

GARCÍA, B. M.; COELLO, C. A. C. An approach based on the use of the ant system to design combinational logic circuits. **Mathware and Soft Computing**, v. 9, n. 2-3, p. 235–250, 2002.

GARCIA, S. *et al.* A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. **IEEE transactions on Knowledge and Data Engineering**, IEEE, v. 25, n. 4, p. 734–750, 2012.

GIACOMANTONIO, C. E.; GOODHILL, G. J. A boolean model of the gene regulatory network underlying mammalian cortical area development. **PLoS Comput Biol**, Public Library of Science, v. 6, n. 9, p. e1000936, 2010.

GIBIM, G. F. B. **Cálculo Diferencial e Integral I**. 1. ed. [S.l.]: Londrina: Editora e Distribuidora Educacional, 2015. (1, v. 1).

GLASS, L.; KAUFFMAN, S. A. The logical analysis of continuous, non-linear biochemical control networks. **Journal of theoretical Biology**, Elsevier, v. 39, n. 1, p. 103–129, 1973.

GOLDBETER, A. A model for circadian oscillations in the drosophila period protein (PER). **Proceedings of the Royal Society of London. Series B: Biological Sciences**, The Royal Society London, v. 261, n. 1362, p. 319–324, 1995.

GOLDMAN, B. W.; PUNCH, W. F. Reducing wasted evaluations in cartesian genetic programming. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2013. p. 61–72.

GOLDMAN, B. W.; PUNCH, W. F. Analysis of cartesian genetic programming's evolutionary mechanisms. **IEEE Trans. on Evolutionary Computation**, IEEE, v. 19, n. 3, p. 359–373, 2015.

HAGHVERDI, L. *et al.* Diffusion pseudotime robustly reconstructs lineage branching. **Nature Methods**, Nature Publishing Group, v. 13, n. 10, p. 845, 2016.

HAMAHASHI, S.; KITANO, H. Simulation of drosophila embryogenesis. In: **Proc. 6th Int. Conf. Artif. Life (Artificial Life VI)**. [S.l.: s.n.], 1998. p. 151–160.

HAN, H. *et al.* Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. **Nucleic acids research**, Oxford University Press, v. 46, n. D1, p. D380–D386, 2018.

HAN, J.; KAMBER, M.; PEI, J. Data mining concepts and techniques third edition. **University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University**, 2012.

HAN, J.; KAMBR, M. **Han J and Kamber M. Data Mining: Concepts and Techniques**. [S.l.]: Los Altos, CA, USA: Morgan Kaufmann Publishers, 2001.

HASE, T. *et al.* Harnessing diversity towards the reconstructing of large scale gene regulatory networks. **PLoS computational biology**, Public Library of Science San Francisco, USA, v. 9, n. 11, p. e1003361, 2013.

HAUSKRECHT, M. *et al.* Feature selection and dimensionality reduction in genomics and proteomics. **Fundamentals of data mining in genomics and proteomics**, Springer, p. 149–172, 2007.

HAYASHI, T. *et al.* Single-cell full-length total rna sequencing uncovers dynamics of recursive splicing and enhancer rnas. **Nature communications**, Nature Publishing Group UK London, v. 9, n. 1, p. 619, 2018.

HODAN, D.; MRAZEK, V.; VASICEK, Z. Semantically-oriented mutation operator in cartesian genetic programming for evolutionary circuit design. In: **Proc. of the 2020 Genetic and Evolutionary Computation Conference**. [S.l.: s.n.], 2020. p. 940–948.

HUSA, J.; KALKREUTH, R. A comparative study on crossover in cartesian genetic programming. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2018. p. 203–219.

HUYNH-THU, V. A. *et al.* Inferring regulatory networks from expression data using tree-based methods. **PloS one**, Public Library of Science San Francisco, USA, v. 5, n. 9, p. e12776, 2010.

HWANG, B.; LEE, J. H.; BANG, D. Single-cell rna sequencing technologies and bioinformatics pipelines. **Experimental & Molecular Medicine**, Nature Publishing Group, v. 50, n. 8, p. 1–14, 2018.

IRRTHUM, A. *et al.* Inferring regulatory networks from expression data using tree-based methods. **PloS One**, Public Library of Science, v. 5, n. 9, p. e12776, 2010.

JACKSON, C. A. *et al.* Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments. **Elife**, eLife Sciences Publications Limited, v. 9, p. e51254, 2020.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. **IEEE Transactions on pattern analysis and machine intelligence**, Ieee, v. 22, n. 1, p. 4–37, 2000.

JI, L.; TAN, K.-L. Mining gene expression data for positive and negative co-regulated gene clusters. **Bioinformatics**, Oxford University Press, v. 20, n. 16, p. 2711–2718, 2004.

JONG, H. D. Modeling and simulation of genetic regulatory systems: a literature review. **J. Comput. Biology**, Mary Ann Liebert, Inc., v. 9, n. 1, p. 67–103, 2002.

KALKREUTH, R.; RUDOLPH, G.; DROSCHINSKY, A. A new subgraph crossover for cartesian genetic programming. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2017. p. 294–310.

KANEHISA, M.; GOTO, S. Kegg: kyoto encyclopedia of genes and genomes. **Nucleic acids research**, Oxford University Press, v. 28, n. 1, p. 27–30, 2000.

KARLEBACH, G.; SHAMIR, R. Modelling and analysis of gene regulatory networks. **Nature reviews Molecular cell biology**, Nature Publishing Group, v. 9, n. 10, p. 770–780, 2008.

KARP, P. D. *et al.* Eco cyc: encyclopedia of escherichia coli genes and metabolism. **Nucleic Acids Research**, Oxford University Press, v. 27, n. 1, p. 55–58, 1999.

KAUFFMAN, S. A. *et al.* **The origins of order: Self-organization and selection in evolution**. [S.l.]: Oxford University Press, USA, 1993.

KHARCHENKO, P. V.; SILBERSTEIN, L.; SCADDEN, D. T. Bayesian approach to single-cell differential expression analysis. **Nature methods**, Nature Publishing Group, v. 11, n. 7, p. 740–742, 2014.

KIKUCHI, S. *et al.* Dynamic modeling of genetic networks using genetic algorithm and s-system. **Bioinformatics**, Oxford University Press, v. 19, n. 5, p. 643–650, 2003.

KIM, S. ppcor: an r package for a fast calculation to semi-partial correlation coefficients. **Commun. Stat. Appl. Methods**, NIH Public Access, v. 22, n. 6, p. 665, 2015.

KIMURA, S. *et al.* Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. **Bioinformatics**, Oxford University Press, v. 21, n. 7, p. 1154–1163, 2005.

KINOSHITA, S.-i.; YAMADA, H. S. Role of self-loop in cell-cycle network of budding yeast. **arXiv preprint arXiv:1811.03162**, 2018.

KITANO, H. **Foundations of systems biology**. [S.l.]: The MIT Press Cambridge, Massachusetts London, England, 2001.

KITANO, H. *et al.* The virtual biology laboratories: A new approach to computational biology. In: MIT PRESS CAMBRIDGE, MA. **Proceedings of the Fourth European Conference on Artificial Life**. [S.l.], 1997. p. 274–283.

KLIPP, E. *et al.* **Systems biology: a textbook**. [S.l.]: John Wiley & Sons, 2016.

KNUDSEN, S. **Guide to analysis of DNA microarray data**. [S.l.]: John Wiley & Sons, 2005.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial intelligence**, Elsevier, v. 97, n. 1-2, p. 273–324, 1997.

KORTHAUER, K. D. *et al.* A statistical approach for identifying differential distributions in single-cell rna-seq experiments. **Genome biology**, Springer, v. 17, p. 1–15, 2016.

KOZA, J. R. **Genetic Programming II, Automatic Discovery of Reusable Subprograms**. [S.l.]: MIT Press, Cambridge, MA, 1992.

KOZA, J. R. Genetic programming as a means for programming computers by natural selection. **Statistics and computing**, Springer, v. 4, n. 2, p. 87–112, 1994.

KRISHNA, K.; MURTY, M. N. Genetic k-means algorithm. **IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 29, n. 3, p. 433–439, 1999.

KRUMSIEK, J. *et al.* Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. **PloS One**, Public Library of Science, v. 6, n. 8, p. e22649, 2011.

KRUMSIEK, J.; WITTMANN, D. M.; THEIS, F. J. From discrete to continuous gene regulation models–a tutorial using the Odefy Toolbox. **Ap. of MATLAB in Science and Eng.**, BoD–Books on Demand, p. 35, 2011.

KUMAR, G.; KUMAR, K. The use of artificial-intelligence-based ensembles for intrusion detection: a review. **Applied Computational Intelligence and Soft Computing**, Hindawi Limited London, UK, United Kingdom, v. 2012, p. 21–21, 2012.

KUTALIK, Z.; TUCKER, W.; MOULTON, V. S-system parameter estimation for noisy metabolic profiles using newton-flow analysis. **IET Systems Biology**, IET, v. 1, n. 3, p. 174–180, 2007.

LÄHDESMÄKI, H.; SHMULEVICH, I.; YLI-HARJA, O. On learning gene regulatory networks under the boolean network model. **Machine learning**, Springer, v. 52, p. 147–167, 2003.

LASHKARI, D. A. *et al.* Yeast microarrays for genome wide parallel genetic and gene expression analysis. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 94, n. 24, p. 13057–13062, 1997.

LAZAR, C. *et al.* A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 9, n. 4, p. 1106–1119, 2012.

LEI, J. *et al.* An approach of gene regulatory network construction using mixed entropy optimizing context-related likelihood mutual information. **Bioinformatics**, Oxford University Press, v. 39, n. 1, p. btac717, 2023.

LELOUP, J.-C.; GOLDBETER, A. A model for circadian rhythms in drosophila incorporating the formation of a complex between the per and tim proteins. **Journal of biological rhythms**, Sage Publications Sage CA: Thousand Oaks, CA, v. 13, n. 1, p. 70–87, 1998.

LI, L. *et al.* Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. **Bioinformatics**, Oxford University Press, v. 17, n. 12, p. 1131–1142, 2001.

LI, X. *et al.* Discovery of time-delayed gene regulatory networks based on temporal gene expression profiling. **BMC bioinformatics**, Springer, v. 7, n. 1, p. 1–20, 2006.

LI, Y. *et al.* Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks. **BMC Bioinformatics**, BioMed Central, v. 11, n. 1, p. 520, 2010.

LIANG, L. *et al.* Selection and validation of reference genes for gene expression studies in codonopsis pilosula based on transcriptome sequence data. **Scientific reports**, Nature Publishing Group, v. 10, n. 1, p. 1–13, 2020.

LIANG, S.; FUHRMAN, S.; SOMOGYI, R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In: **Biocomputing**. [S.l.: s.n.], 1998. v. 3.

LIM, C. Y. *et al.* Btr: training asynchronous boolean models using single-cell expression data. **BMC bioinformatics**, BioMed Central, v. 17, n. 1, p. 1–18, 2016.

LIN, Y. *et al.* scclassify: sample size estimation and multiscale classification of cells using single and multiple reference. **Molecular systems biology**, v. 16, n. 6, p. e9389, 2020.

LIU, S.; TRAPNELL, C. Single-cell transcriptome sequencing: recent advances and remaining challenges. **F1000Research**, Faculty of 1000 Ltd, v. 5, 2016.

LIU, Z.-P. *et al.* Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. **Database**, Oxford University Press, v. 2015, p. bav095, 2015.

LONARDI, S.; SZPANKOWSKI, W.; YANG, Q. Finding biclusters by random projections. In: SPRINGER. **Annual Symposium on Combinatorial Pattern Matching**. [S.l.], 2004. p. 102–116.

LOVRICS, A. *et al.* Boolean modelling reveals new regulatory connections between transcription factors orchestrating the development of the ventral spinal cord. **PloS One**, Public Library of Science, v. 9, n. 11, p. e111430, 2014.

LUO, J. *et al.* Gene regulatory network analysis identifies key genes and regulatory mechanisms involved in acute myocardial infarction using bulk and single cell rna-seq data. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2021–08, 2021.

MA, B. *et al.* Identification of gene regulatory networks by integrating genetic programming with particle filtering. **IEEE Access**, IEEE, v. 7, p. 113760–113770, 2019.

MA, W. *et al.* Defining network topologies that can achieve biochemical adaptation. **Cell**, Elsevier, v. 138, n. 4, p. 760–773, 2009.

MACQUEEN, J. *et al.* Some methods for classification and analysis of multivariate observations. In: **Proc. of the Berkeley symposium on mathematical statistics and probability**. [S.l.: s.n.], 1967. v. 1, p. 281–297.

MADEIRA, S. C.; OLIVEIRA, A. L. An evaluation of discretization methods for non-supervised analysis of time-series gene expression data. **INESC-ID Technical Report**, Citeseer, v. 42, p. 2005, 2005.

MADEIRA, S. C. *et al.* Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 7, n. 1, p. 153–165, 2008.

MAHANTA, P. *et al.* Discretization in gene expression data analysis: a selected survey. In: **Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology**. [S.l.: s.n.], 2012. p. 69–75.

MANFRINI, F. A. L.; BERNARDINO, H. S.; BARBOSA, H. J. C. A novel efficient mutation for evolutionary design of combinational logic circuits. In: SPRINGER. **Intl. Conf. on Parallel Problem Solving from Nature**. [S.l.], 2016. p. 665–674.

MANFRINI, F. A. L. *et al.* Estratégias de busca no projeto evolucionista de circuitos combinacionais. Universidade Federal de Juiz de Fora (UFJF), 2017.

MARBACH, D. *et al.* Revealing strengths and weaknesses of methods for gene network inference. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 107, n. 14, p. 6286–6291, 2010.

MARIONI, J. C. *et al.* Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. **Genome research**, Cold Spring Harbor Lab, v. 18, n. 9, p. 1509–1517, 2008.

MARKO, N. F.; WEIL, R. J. Non-gaussian distributions affect identification of expression patterns, functional annotation, and prospective classification in human cancer genomes. **PLoS One**, Public Library of Science San Francisco, USA, v. 7, n. 10, p. e46935, 2012.

MATSUMOTO, H. *et al.* Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. **Bioinformatics**, Oxford University Press, v. 33, n. 15, p. 2314–2321, 2017.

MCADAMS, H. H.; SHAPIRO, L. Circuit simulation of genetic networks. **Science**, American Association for the Advancement of Science, v. 269, n. 5224, p. 650–656, 1995.

MCCALL, M. N. Estimation of gene regulatory networks. **Postdoc journal: a journal of postdoctoral research and postdoctoral affairs**, NIH Public Access, v. 1, n. 1, p. 60, 2013.

MCCLUSKEY, E. J. Minimization of boolean functions. **Bell Labs Technical Journal**, Wiley Online Library, v. 35, n. 6, p. 1417–1444, 1956.

MEZURA-MONTES, E.; COELLO, C. A. C. Constraint-handling in nature-inspired numerical optimization: Past, present and future. **Swarm and Evolutionary Computation**, v. 1, n. 4, p. 173–194, 2011.

MICHAELS, G. S. *et al.* Cluster analysis and data visualization of large-scale gene expression data. In: **Pacific symposium on biocomputing**. [S.l.: s.n.], 1998. v. 3, p. 42–53.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)**. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN 3-540-60676-9.

MILLER, J. F. Cartesian genetic programming. **CGP**, Springer, p. 17–34, 2011.

MILLER, J. F.; THOMSON, P.; FOGARTY, T. **Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study**. [S.l.]: Wiley, 1997.

MOERMAN, T. *et al.* GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. **Bioinformatics**, Oxford University Press, v. 35, n. 12, p. 2159–2161, October 2019.

MOIGNARD, V. *et al.* Decoding the regulatory network of early blood development from single-cell gene expression measurements. **Nature Biotechnology**, Nature Publishing Group, v. 33, n. 3, p. 269–276, 2015.

MÖLLER-LEVET, C. S.; CHU, K.; WOLKENHAUER, O. Dna microarray data clustering based on temporal variation: Fcv with tsd preclustering. **Applied Bioinformatics**, v. 2, n. 1, p. 35–45, 2003.

MONTAGNA, S.; BRACCINI, M.; ROLI, A. The impact of self-loops on boolean networks attractor landscape and implications for cell differentiation modelling. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 18, n. 6, p. 2702–2713, 2020.

MÜNZ, G.; LI, S.; CARLE, G. Traffic anomaly detection using k-means clustering. In: **Gi/itg workshop mmbnet**. [S.l.: s.n.], 2007. v. 7, n. 9.

NESTOROWA, S. *et al.* A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. **Blood, The Journal of the American Society of Hematology**, American Society of Hematology Washington, DC, v. 128, n. 8, p. e20–e31, 2016.

NG, H. *et al.* Medical image segmentation using k-means clustering and improved watershed algorithm. In: IEEE. **2006 IEEE southwest symposium on image analysis and interpretation**. [S.l.], 2006. p. 61–65.

NOMAN, N.; IBA, H. Inference of gene regulatory networks using s-system and differential evolution. In: **Proceedings of the 7th annual conference on genetic and evolutionary computation**. [S.l.: s.n.], 2005. p. 439–446.

NORDICK, B.; HONG, T. Identification, visualization, statistical analysis and mathematical modeling of high-feedback loops in gene regulatory networks. **BMC bioinformatics**, BioMed Central, v. 22, n. 1, p. 1–21, 2021.

OCONE, A. *et al.* Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data. **Bioinformatics**, Oxford University Press, v. 31, n. 12, p. i89–i96, 2015.

PALSSON, B. **Systems biology**. [S.l.]: Cambridge university press, 2015.

PASSARGE, E. **Genetica texto y atlas**. [S.l.]: Ed. Médica Panamericana, 2009.

PENFOLD, C. A.; WILD, D. L. How to infer gene networks from expression profiles, revisited. **Interface focus**, The Royal Society, v. 1, n. 6, p. 857–870, 2011.

PENSA, R. G. *et al.* Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: SPRINGER. **Proc. of the Intl. Conf. on Data Mining in Bioinformatics**. [S.l.], 2004. p. 24–30.

PETRATOU, K. *et al.* A systems biology approach uncovers the core gene regulatory network governing iridophore fate choice from the neural crest. **PLoS genetics**, Public Library of Science San Francisco, CA USA, v. 14, n. 10, p. e1007402, 2018.

PFEUTY, B.; KANEKO, K. The combination of positive and negative feedback loops confers exquisite flexibility to biochemical switches. **Physical biology**, IOP Publishing, v. 6, n. 4, p. 046013, 2009.

PIRGAZI, J.; KHANTEYMOORI, A. R. A robust gene regulatory network inference method base on kalman filter and linear regression. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 7, p. e0200094, 2018.

POKHILKO, A. *et al.* The clock gene circuit in arabidopsis includes a repressilator with additional feedback loops. **Molecular systems biology**, John Wiley & Sons, Ltd Chichester, UK, v. 8, n. 1, p. 574, 2012.

PONZONI, I. *et al.* Inferring adaptive regulation thresholds and association rules from gene expression data through combinatorial optimization learning. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 4, n. 4, p. 624–634, 2007.

PRACHEDES, L. *et al.* Inferência de redes de regulação gênica usando programação genética cartesiana paralela. In: . SBC, 2022b. p. 74–79. ISSN 2763-8987. Disponível em: <https://sol.sbc.org.br/index.php/sbcas_estendido/article/view/20504>.

PRACHEDES, L. N. S. *et al.* High-performance cartesian genetic programming on gpu for the inference of gene regulatory networks using scrna-seq time-series data. In: **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.: s.n.], 2022a. p. 2063–2070.

PRATAPA, A. *et al.* Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. **Nature Methods**, Nature Publishing Group, v. 17, n. 2, p. 147–154, 2020.

PREDEU, H. T. A. **Analysis of single cell RNA-Seq Data**. Jul 2022. Acessado em 23 de Julho de 2022. Disponível em: <https://www.singlecellcourse.org/introduction-to-single-cell-rna-seq.html>.

PRILL, R. J. *et al.* Towards a rigorous assessment of systems biology models: the dream3 challenges. **PloS one**, Public Library of Science San Francisco, USA, v. 5, n. 2, p. e9202, 2010.

PUŠNIK, Ž. *et al.* Review and assessment of boolean approaches for inference of gene regulatory networks. **Heliyon**, Elsevier, 2022.

QIU, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. **Nature Methods**, Nature Publishing Group, v. 14, n. 10, p. 979, 2017.

QU, J. *et al.* A novel discretization method for processing digital gene expression profiles. In: IEEE. **2013 7th Intl. Conf. on Systems Biology (ISB)**. [S.l.], 2013. p. 134–138.

REINITZ, J.; MJOLSNESS, E.; SHARP, D. H. Model for cooperative control of positional information in drosophila by bicoid and maternal hunchback. **Journal of Experimental Zoology**, Wiley Online Library, v. 271, n. 1, p. 47–56, 1995.

RICHELDI, M.; ROSSOTTO, M. Class-driven statistical discretization of continuous attributes. In: SPRINGER. **European Conference on Machine Learning**. [S.l.], 1995. p. 335–338.

RÍOS, O. *et al.* A boolean network model of human gonadal sex determination. **Theor. Biol. Med. Model.**, BioMed Central, v. 12, n. 1, p. 1–18, 2015.

RITCHIE, M. E. *et al.* limma powers differential expression analyses for rna-sequencing and microarray studies. **Nucleic acids research**, Oxford Academic, v. 43, n. 7, p. e47–e47, 2015.

ROKACH, L.; MAIMON, O. Clustering methods. **Data mining and knowledge discovery handbook**, Springer, p. 321–352, 2005.

RONEN, M. *et al.* Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 99, n. 16, p. 10555–10560, 2002.

ROZENBERG, G.; BÄCK, T.; KOK, J. N. **Handbook of natural computing**. [S.l.]: Springer, 2012.

SAELENS, W. *et al.* A comparison of single-cell trajectory inference methods. **Nature biotechnology**, Nature Publishing Group, v. 37, n. 5, p. 547–554, 2019.

SAEYS, Y.; INZA, I.; LARRANAGA, P. A review of feature selection techniques in bioinformatics. **bioinformatics**, Oxford University Press, v. 23, n. 19, p. 2507–2517, 2007.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PloS one**, Public Library of Science San Francisco, CA USA, v. 10, n. 3, p. e0118432, 2015.

SANCHEZ-CASTILLO, M. *et al.* A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. **Bioinformatics**, Oxford University Press, v. 34, n. 6, p. 964–970, 2018.

SANGUINETTI, G. *et al.* Gene regulatory network inference: an introductory survey. In: **Gene Regulatory Networks**. [S.l.]: Springer, 2019. p. 1–23.

SAURO, H. M. **Essentials of Biochemical Modeling**. 1. ed. [S.l.]: Ambrosius Publishing, 2014. ISBN 978-0982477328.

SAVAGEAU, M. Rules for the evolution of gene circuitry. In: WORLD SCIENTIFIC MAUI, HAWAII. **Pacific Symposium on Biocomputing**. [S.l.], 1998. v. 3, p. 54–65.

SCHAFFTER, T.; MARBACH, D.; FLOREANO, D. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. **Bioinformatics**, Oxford University Press, v. 27, n. 16, p. 2263–2270, 2011.

SCHMIDT, M. D.; LIPSON, H. Data-mining dynamical systems: Automated symbolic system identification for exploratory analysis. In: **Engineering Systems Design and Analysis**. [S.l.: s.n.], 2008. v. 48364, p. 643–649.

SCHRYNEMACKERS, M.; KÜFFNER, R.; GEURTS, P. On protocols and measures for the validation of supervised methods for the inference of biological networks. **Frontiers in genetics**, Frontiers Media SA, v. 4, p. 262, 2013.

SCHUH, L. *et al.* Gene networks with transcriptional bursting recapitulate rare transient coordinated high expression states in cancer. **Cell systems**, Elsevier, v. 10, n. 4, p. 363–378, 2020.

SCHWEFEL, H.-P. **Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie**. [S.l.]: Springer, 1977.

SETTY, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. **Nat. Biotechnol.**, Nature Publishing Group, v. 34, n. 6, p. 637–645, 2016.

SHALEK, A. K. *et al.* Single-cell rna-seq reveals dynamic paracrine control of cellular variation. **Nature**, Nature Publishing Group UK London, v. 510, n. 7505, p. 363–369, 2014.

SHEN-ORR, S. S. *et al.* Network motifs in the transcriptional regulation network of escherichia coli. **Nature genetics**, Nature Publishing Group, v. 31, n. 1, p. 64–68, 2002.

SHI, N. *et al.* Aten: And/or tree ensemble for inferring accurate boolean network topology and dynamics. **Bioinformatics**, Oxford University Press, v. 36, n. 2, p. 578–585, 2020.

SHIROTA, M.; KINOSHITA, K. Discrepancies between human dna, mrna and protein reference sequences and their relation to single nucleotide variants in the human population. **Database**, Oxford Academic, v. 2016, 2016.

SILVA, B. M.; BERNARDINO, H. S.; BARBOSA, H. J. Human activity recognition using parallel cartesian genetic programming. In: **2021 IEEE Congress on Evolutionary Computation (CEC)**. Kraków, Poland: IEEE, 2021. p. 474–481.

SILVA, J. E. da; BERNARDINO, H. Cartesian genetic programming with crossover for designing combinational logic circuits. In: IEEE. **2018 7th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2018. p. 145–150.

SILVA, J. E. da; SOUZA, L. A. M. de; BERNARDINO, H. Cartesian genetic programming with guided and single active mutations for designing combinational logic circuits. In: SPRINGER. **Proc. of the 5th Conference on machine Learning, Optimization and Data science (LOD)**. [S.l.], 2019. p. 1–12.

SILVA, J. E. H. d. *et al.* On the analysis of cgp mutation operators when inferring gene regulatory networks using scrna-seq time series data. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2021. p. 264–279.

SILVA, J. E. H. d. *et al.* **On the Discretization Methods for Single-Cell RNA-Sequencing Data when Inferring Gene Regulatory Networks via Cartesian Genetic Programming**. ABMEC, 2021. Disponível em: <https://cilamce.com.br/anais/arearestrita/apresentacoes/219/9668.pdf>.

SILVA, J. E. H. d. *et al.* Inference of gene regulatory networks from single-cell rna-sequencing data using cartesian genetic programming. **Applied Soft Computing**, Elsevier, 2023.

SILVA, J. E. H. da *et al.* Inferring gene regulatory network models from time-series data using metaheuristics. In: IEEE. **2020 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2020. p. 1–8.

SILVA, J. E. H. da *et al.* A survey of the methodological process of modeling, inference, and evaluation of gene regulatory networks using scrna-seq data. **Biosystems**, Elsevier, p. 105126, 2024.

SILVA, J. E. H. da *et al.* Inferring gene regulatory networks from single-cell rna-sequencing experimental data using cartesian genetic programming. In: IEEE. **2024 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2024. p. 1–8.

SILVA, J. E. H. da *et al.* A data-distribution and successive spline points based discretization approach for evolving gene regulatory networks from scrna-seq time-series data using cartesian genetic programming. **Biosystems**, Elsevier, p. 105126, 2024.

SILVA, J. E. H. da *et al.* On the use of clonal selection principle in cartesian genetic programming for designing combinational logic circuits. In: SBC. **Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2022. p. 152–163.

SILVER, N. *et al.* Selection of housekeeping genes for gene expression studies in human reticulocytes using real-time pcr. **BMC molecular biology**, BioMed Central, v. 7, n. 1, p. 1–9, 2006.

SINHA, S. *et al.* Behavior-related gene regulatory networks: A new level of organization in the brain. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 117, n. 38, p. 23270–23279, 2020.

SMET, R. D.; MARCHAL, K. Advantages and limitations of current network inference methods. **Nature Reviews Microbiology**, Nature Publishing Group UK London, v. 8, n. 10, p. 717–729, 2010.

SMITH, J.; THEODORIS, C.; DAVIDSON, E. H. A gene regulatory network subcircuit drives a dynamic pattern of gene expression. **Science**, American Association for the Advancement of Science, v. 318, n. 5851, p. 794–797, 2007.

SOINOV, L. A.; KRESTYANINOVA, M. A.; BRAZMA, A. Towards reconstruction of gene networks from expression data by supervised learning. **Genome biology**, BioMed Central, v. 4, n. 1, p. 1–10, 2003.

SONESON, C.; ROBINSON, M. D. Bias, robustness and scalability in single-cell differential expression analysis. **Nature methods**, Nature Publishing Group US New York, v. 15, n. 4, p. 255–261, 2018.

SOUZA, L. A. M. de *et al.* A benchmark suite for designing combinational logic circuits via metaheuristics. **Applied Soft Computing**, Elsevier, v. 91, p. 106246, 2020.

SPECHT, A. T.; LI, J. LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. **Bioinformatics**, Oxford University Press, v. 33, n. 5, p. 764–766, 2017.

SPELLMAN, P. T. *et al.* Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. **Molecular biology of the cell**, Am Soc Cell Biol, v. 9, n. 12, p. 3273–3297, 1998.

SPIETH, C. *et al.* A memetic inference method for gene regulatory networks based on s-systems. In: IEEE. **Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)**. [S.l.], 2004. v. 1, p. 152–157.

STREET, K. *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. **BMC Genomics**, Springer, v. 19, n. 1, p. 1–16, 2018.

STREICHERT, F. *et al.* Comparing genetic programming and evolution strategies on inferring gene regulatory networks. In: **Genetic and Evolutionary Computation Conference**. Berlin, Heidelberg: Springer, 2004. p. 471–480.

SU, K.; YU, T.; WU, H. Accurate feature selection improves single-cell rna-seq cell clustering. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 5, p. bbab034, 2021.

SUDHOLT, D. Parallel evolutionary algorithms. **Springer Handbook of Computational Intelligence**, Springer, p. 929–959, 2015.

SZKLARCZYK, D. *et al.* String v10: protein–protein interaction networks, integrated over the tree of life. **Nucleic acids research**, Oxford University Press, v. 43, n. D1, p. D447–D452, 2015.

TARCA, A. L.; ROMERO, R.; DRAGHICI, S. Analysis of microarray experiments of gene expression profiling. **American journal of obstetrics and gynecology**, Elsevier, v. 195, n. 2, p. 373–388, 2006.

THOMAS, R. *et al.* A model-based optimization framework for the inference on gene regulatory networks from dna array data. **Bioinformatics**, Oxford University Press, v. 20, n. 17, p. 3221–3235, 2004.

TIAN, L. *et al.* scrna-seq mixology: towards better benchmarking of single cell rna-seq analysis methods. **BioRxiv**, Cold Spring Harbor Laboratory, p. 433102, 2019.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas digitais: princípios e aplicações**. [S.l.]: Prentice Hall, 2003.

TORRENTÉ, L. de *et al.* The shape of gene expression distributions matter: how incorporating distribution shape improves the interpretation of cancer transcriptomic data. **BMC bioinformatics**, BioMed Central, v. 21, n. 21, p. 1–18, 2020.

TRAPNELL, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. **Nat. Biotechnol.**, Nature Publishing Group, v. 32, n. 4, p. 381, 2014.

TURNER, A. J.; MILLER, J. F. Neutral genetic drift: an investigation using cartesian genetic programming. **Genetic Programming and Evolvable Machines**, Springer, v. 16, n. 4, p. 531–558, 2015.

VANS, E.; PATIL, A.; SHARMA, A. Feats: feature selection-based clustering of single-cell rna-seq data. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 4, p. bbaa306, 2021.

VASICEK, Z. Cartesian GP in optimization of combinational circuits with hundreds of inputs and thousands of gates. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2015. p. 139–150.

VASICEK, Z. Bridging the gap between evolvable hardware and industry using cartesian genetic programming. In: **Inspired by Nature**. [S.l.]: Springer, 2018. p. 39–55.

VASICEK, Z.; SEKANINA, L. Circuit approximation using single-and multi-objective cartesian GP. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2015. p. 217–229.

VICKERS, N. J. Animal communication: when i'm calling you, will you answer too? **Current biology**, Elsevier, v. 27, n. 14, p. R713–R715, 2017.

VOIT, E. O. **Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists**. [S.l.]: Cambridge University Press, 2000.

WALKER, J. A.; MILLER, J. F.; CAVILL, R. A multi-chromosome approach to standard and embedded cartesian genetic programming. In: ACM. **Proc. of the 8th annual conference on genetic and evolutionary computation**. [S.l.], 2006. p. 903–910.

WANG, H.; QIAN, L.; DOUGHERTY, E. Inference of gene regulatory networks using s-system: a unified approach. **IET systems biology**, IET, v. 4, n. 2, p. 145–156, 2010.

WANG, R.-S.; SAADATPOUR, A.; ALBERT, R. Boolean modeling in systems biology: an overview of methodology and applications. **Physical Biology**, IOP Publishing, v. 9, n. 5, p. 055001, 2012.

WANG, Z.; GERSTEIN, M.; SNYDER, M. RNA-Seq: a revolutionary tool for transcriptomics. **Nature Reviews Genetics**, Nature Publishing Group, v. 10, n. 1, p. 57, 2009.

WATSON, J. D. *et al.* **Biologia molecular do gene**. [S.l.]: Artmed Editora, 2015.

WATSON, J. D.; CRICK, F. *et al.* A structure for deoxyribose nucleic acid. Macmillan, 1953.

WEAVER, R. **EBOOK: Molecular Biology**. [S.l.]: McGraw Hill, 2011.

WILLIAMS, P. L.; BEER, R. D. Nonnegative decomposition of multivariate information. **arXiv preprint arXiv:1004.2515**, 2010.

WITTMANN, D. M. *et al.* Transforming boolean models to continuous models: methodology and application to t-cell receptor signaling. **BMC Systems Biology**, BioMed Central, v. 3, n. 1, p. 98, 2009.

WOODHOUSE, S. *et al.* Scns: a graphical tool for reconstructing executable regulatory networks from single-cell genomic data. **BMC systems Biology**, BioMed Central, v. 12, n. 1, p. 1–7, 2018.

YANG, B. *et al.* Hscvfnt: Inference of time-delayed gene regulatory network based on complex-valued flexible neural tree model. **International Journal of Molecular Sciences**, MDPI, v. 19, n. 10, p. 3178, 2018.

YANG, P.; HUANG, H.; LIU, C. Feature selection revisited in the single-cell era. **Genome Biology**, Springer, v. 22, p. 1–17, 2021.

YANG, P. *et al.* A particle swarm based hybrid system for imbalanced medical data sampling. In: SPRINGER. **BMC genomics**. [S.l.], 2009. v. 10, p. 1–14.

YAO, G. *et al.* A bistable rb–e2f switch underlies the restriction point. **Nature cell biology**, Nature Publishing Group UK London, v. 10, n. 4, p. 476–482, 2008.

YIP, S. H.; SHAM, P. C.; WANG, J. Evaluation of tools for highly variable gene discovery from single-cell rna-seq data. **Briefings in bioinformatics**, Oxford University Press, v. 20, n. 4, p. 1583–1589, 2019.

ZHANG, S. *et al.* Inference of cell type-specific gene regulatory networks on cell lineages from single cell omic datasets. **Nature Communications**, Nature Publishing Group UK London, v. 14, n. 1, p. 3064, 2023.

ZHAO, M. *et al.* A comprehensive overview and critical evaluation of gene regulatory network inference technologies. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 5, p. bbab009, 2021.

ZIEGENHAIN, C. *et al.* Comparative analysis of single-cell rna sequencing methods. **Molecular cell**, Elsevier, v. 65, n. 4, p. 631–643, 2017.

ZOMAYA, A. Y.; ELLOUMI, M. **Biological knowledge discovery handbook: Preprocessing, mining and postprocessing of biological data**. [S.l.]: John Wiley & Sons, 2013.

APÊNDICE A – Resultados Tabulares da Qualidade do Agrupamento

Tabela 48 – hESC - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
			500nTF			
ChIP-Seq	0 1	36.87% 65.24%	0.73795	0.71499	0.72629	62%
NonSpecific	0 1	15.79% 11.74%	0.51117	0.35257	0.41731	12.2%
STRING	0 1	15.79% 18.74%	0.30497	0.23792	0.26730	18.4%
			500TF			
ChIP-Seq	0 1	59.65% 89.84%	0.30197	0.30565	0.30800	86.4%
NonSpecific	0 1	61.4% 77.88%	0.00749	0.00821	0.00783	76%
STRING	0 1	35.09% 38.37%	0.004056	0.005814	0.00478	38%
			1000nTF			
ChIP-Seq	0 1	70.65% 77.07%	0.87629	0.87014	0.87320	75.3%
NonSpecific	0 1	51.81% 49.31%	0.62201	0.59091	0.60606	50%
STRING	0 1	27.17% 20.58%	0.30605	0.28948	0.29753	22.4%
			1000TF			
ChIP-Seq	0 1	86.85% 90.37%	0.82063	0.80689	0.81370	89.36%
NonSpecific	0 1	82.13% 81.23%	0.32575	0.30926	0.31729	81.49%
STRING	0 1	57.57% 47.37%	0.12623	0.11853	0.12226	50.28%

Tabela 49 – hHep - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
500nTF						
ChIP-Seq	0	80.47%	0.94592	0.95454	0.95027	75.6%
	1	64.97%				
NonSpecific	0	20.41%	0.49169	0.52658	0.508537	17.6%
	1	11.46%				
STRING	0	38.78%	0.61263	0.627067	0.619762	32.2%
	1	17.83%				
500TF						
ChIP-Seq	0	95.65%	0.72091	0.67159	0.69538	92.19%
	1	95.45%				
	2	91.14%				
	3	93.15%				
	4	89.66%				
NonSpecific	0	91.3%	0.265731	0.24107	0.25280	87.76
	1	72.73%				
	2	81.01%				
	3	86.72%				
	4	94.83%				
STRING	0	75.36%	0.09948	0.08452	0.09139	69.2%
	1	59.09%				
	2	39.24%				
	3	68.52%				
	4	90.09%				
1000nTF						
ChIP-Seq	0	74.71%	0.93933	0.92483	0.93202	81.6%
	1	83.98%				
NonSpecific	0	17.9%	0.60770	0.55700	0.58125	21%
	1	22.07%				
STRING	0	15.18%	0.61113	0.56859	0.58909	34.2%
	1	40.78%				
1000TF						
ChIP-Seq	0	92.58%	0.78745	0.78260	0.78501	91.92%
	1	88.15%				
	2	94.12%				
	3	91.24%				
NonSpecific	0	82.74%	0.32664	0.32261	0.32462	84.53
	1	94.79%				
	2	86.27%				
	3	80.08%				
STRING	0	60.87%	0.12396	0.11570	0.11969	61.40%
	1	91.00%				
	2	63.24%				
	3	36.65%				

Tabela 50 – mDC - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
500nTF						
ChIP-Seq	0	6.97%	0.73923	0.69956	0.71885	6.8%
	1	6.47%				
NonSpecific	0	6.36%	0.30870	0.30870	0.30870	6.6%
	1	7.06%				
STRING	0	16.06%	0.61047	0.59951	0.60494	17.4%
	1	20%				
500TF						
ChIP-Seq	0	55.79%	0.75024	0.71724	0.73337	54.57%
	1	50.51%				
	2	53.92%				
	3	75%				
NonSpecific	0	95.79%	0.38478	0.32547	0.35265	78.32%
	1	75.76%				
	2	65.36%				
	3	62.5%				
STRING	0	81.05%	0.27464	0.20788	0.23664	59.32%
	1	59.09%				
	2	39.87%				
	3	53.12%				
1000nTF						
ChIP-Seq	0	26.44%	0.84284	0.80088	0.82132	26.1%
	1	25.1%				
NonSpecific	0	25.9%	0.55527	0.51299	0.53330	27.2%
	1	31.08%				
STRING	0	23.9%	0.53696	0.47856	0.50608	25%
	1	28.29%				
1000TF						
ChIP-Seq	0	52.04%	0.84085	0.79784	0.81878	52.23%
	1	52.85%				
NonSpecific	0	74.53%	0.52541	0.43127	0.47371	74.19%
	1	73.10%				
STRING	0	51.34%	0.39610	0.29902	0.34079	51.55%
	1	52.22%				

Tabela 51 – mESC - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
500nTF						
ChIP-Seq	0	75%	0.52269	0.11614	0.19006	73.4%
	1	73.39%				
NonSpecific	0	25%	0.0	1.0	0.0	41.2%
	1	41.33%				
STRING	0	25%	0.0	1.0	0.0	17.6%
	1	17.54%				
500TF						
ChIP-Seq	0	86.78%	0.53199	0.41385	0.46554	87.23%
	1	87.44%				
	2	80%				
NonSpecific	0	77.97%	0.12037	0.098862	0.10856	80%
	1	80.73%				
	2	80%				
STRING	0	60.68%	0.04453	0.03601	0.03982	56.86%
	1	56.95%				
	2	40%				
1000nTF						
ChIP-Seq	0	69.98%	0.67579	0.62778	0.65090	77.5%
	1	84.21%				
	2	90.91%				
	3	81.69%				
NonSpecific	0	52.61%	0.46925	0.45913	0.46414	49.8%
	1	62.57%				
	2	54.55%				
	3	41.69%				
STRING	0	22.58%	0.26065	0.25691	0.25877	22.5%
	1	32.75%				
	2	27.27%				
	3	18.07%				
1000TF						
ChIP-Seq	0	87.98%	0.61095	0.58328	0.59679	85.49%
	1	88.51%				
	2	82.00%				
NonSpecific	0	78.86%	0.15709	0.14082	0.14851	75.37%
	1	79.89%				
	2	70.39%				
STRING	0	50.07%	0.06518	0.05784	0.06129	49.32%
	1	60.92%				
	2	45.57%				

Tabela 52 – hHSC-E - Qualidade do Agrupamento - GWR (genes com relaçoes regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
			500nTF			
ChIP-Seq	0	93.82%	0.87928	0.87928	0.87928	91.8%
	1	85.94%				
NonSpecific	0	37.9%	0.58920	0.60061	0.59485	33.8%
	1	21.88%				
STRING	0	25.27%	0.48771	0.48771	0.48771	22.4%
	1	21.88%				
			500TF			
ChIP-Seq	0	98.53%	0.74253	0.75669	0.74954	98.15%
	1	97.37%				
NonSpecific	0	62.82%	0.29804	0.30445	0.30121	63.49%
	1	64.91%				
STRING	0	43.7%	0.15657	0.15979	0.15816	42.61%
	1	40.35%				
			1000nTF			
ChIP-Seq	0	94%	0.86999	0.86999	0.86999	96.8%
	1	97.73%				
NonSpecific	0	27.2%	0.54778	0.54025	0.54399	36.1%
	1	39.07%				
STRING	0	14%	0.36155	0.34213	0.35158	24.1%
	1	27.47%				
			1000TF			
ChIP-Seq	0	96.44%	0.79655	0.80487	0.80069	97.76%
	1	98.27%				
NonSpecific	0	55.49%	0.39588	0.40584	0.40080	56.48%
	1	56.86%				
STRING	0	33.53%	0.22618	0.23112	0.22863	35.47%
	1	36.22%				

Tabela 53 – mHSC-GM - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
500nTF						
ChIP-Seq	0	97.31%	0.85589	0.85791	0.85690	96.2%
	1	93.94%				
NonSpecific	0	25.67%	0.56515	0.57768	0.57135	26%
	1	26.67%				
STRING	0	18.81%	0.37842	0.39474	0.38641	17.8%
	1	15.76%				
500TF						
ChIP-Seq	0	98.56%	0.69920	0.70998	0.70455	97.78%
	1	98.85%				
	2	96.83%				
	3	95.83%				
NonSpecific	0	46.04%	0.32111	0.32227	0.321689	47.47%
	1	43.51%				
	2	60.32%				
	3	50%				
STRING	0	34.53%	0.16635	0.16599	0.16617	32.59%
	1	32.82%				
	2	44.44%				
	3	26.19%				
1000nTF						
ChIP-Seq	0	96.57%	0.81860	0.81514	0.81687	95.20%
	1	92.41%				
	2	94.98%				
	3	93.33%				
NonSpecific	0	37.04%	0.49628	0.51159	0.50382	36.00%
	1	35.44%				
	2	40.54%				
	3	27.69%				
STRING	0	31.91%	0.30761	0.31222	0.30990	27.30%
	1	17.72%				
	2	30.50%				
	3	15.90%				
1000TF						
ChIP-Seq	0	95.78%	0.81911	0.82135	0.82023	96.20%
	1	94.88%				
	2	97.07%				
NonSpecific	0	52.41%	0.45106	0.46399	0.45743	46.91%
	1	42.13%				
	2	45.79%				
STRING	0	34.04%	0.23261	0.24064	0.23655	31.54%
	1	22.05%				
	2	34.43%				

Tabela 54 – mHSC-L - Qualidade do Agrupamento - GWR (genes com relações regulatórias).

		GWR (%)	Homogeneidade	Completude	V-Measure	Total GWR (%)
			500nTF			
ChIP-Seq	0	91.41%	0.76613	0.76779	0.76696	93.8%
	1	95.32%				
	2	95.42%				
NonSpecific	0	24.24%	0.36425	0.36415	0.36420	28%
	1	28.65%				
	2	32.82%				
STRING	0	14.14%	0.19828	0.19687	0.19757	12.4%
	1	5.85%				
	2	18.32%				
			500TF			
ChIP-Seq	0	92.35%	0.78851	0.79213	0.79031	93.75%
	1	97.02%				
NonSpecific	0	28.32%	0.36646	0.36486	0.36566	30%
	1	33.93%				
STRING	0	10.71%	0.15373	0.15470	0.15421	13.21%
	1	19.05%				
			1000nTF			
ChIP-Seq	0	91.67%	0.81206	0.81206	0.81206	92.49%
	1	94.68%				
NonSpecific	0	26.79%	0.38713	0.38538	0.38625	28.61%
	1	33.51%				
STRING	0	10.52%	0.17103	0.16780	0.16940	12.43%
	1	17.55%				
			1000TF			
ChIP-Seq	0	91.67%	0.81206	0.81206	0.81206	92.49%
	1	94.68%				
NonSpecific	0	26.79%	0.38713	0.38538	0.38625	28.61%
	1	33.51%				
STRING	0	10.52%	0.17103	0.16780	0.16940	12.43%
	1	17.55%				

ANEXO A – Modelo de Ritmo Circadiano de 5 espécies

Este anexo apresenta o modelo de EDO originais para o ritmo circadiano que considera 5 (GOLDBETER, 1995).

$$\frac{dM}{dt} = v_s \frac{K_1^n}{K_1^n + P_N^n}$$

$$\frac{dP_0}{dt} = k_s M - V_1 \frac{P_0}{K_1 + P_0} + V_2 \frac{P_1}{K_2 + P_1}$$

$$\frac{dP_1}{dt} = V_1 \frac{P_0}{K_1 + P_0} - V_2 \frac{P_1}{K_2 + P_1} - V_3 \frac{P_1}{K_3 + P_1} + V_4 \frac{P_2}{K_4 + P_2}$$

$$\frac{dP_2}{dt} = V_3 \frac{P_1}{K_3 + P_1} - V_4 \frac{P_2}{K_4 + P_2} - k_1 P_2 + k_2 P_N - v_d \frac{P_2}{K_d + P_2}$$

$$\frac{dP_N}{dt} = k_1 P_2 - k_2 P_N$$

A quantidade total (não conservada) da proteína PER, P_t é dada por:

$$P_t = P_0 + P_1 + P_2 + P_N \tag{.1}$$

ANEXO B – Modelo de Ritmo Circadiano de 10 espécies

Este anexo apresenta o modelo de EDO originais para o ritmo circadiano que considera 10 espécies (LELOUP; GOLDBETER, 1998).

$$\frac{dM_p}{dt} = v_{sP}\frac{K_{IP}^n}{K_{IP}^n+C_N^n} - v_{mP}\frac{M_p}{K_{mP}+M_p} - k_dM_p$$

$$\frac{dP_0}{dt} = k_{sP}M_p - V_{1P}\frac{P_0}{K_{1P}+P_0} + V_{2P}\frac{P_1}{K_{2P}+P_1} - k_dP_0$$

$$\frac{dP_1}{dt} = V_{1P}\frac{P_0}{K_{1P}+P_0} - V_{2P}\frac{P_1}{K_{2P}+P_1} - V_{3P}\frac{P_1}{K_{3P}+P_1} + V_{4P}\frac{P_2}{K_{4P}+P_2} - k_dP_1$$

$$\frac{dP_2}{dt} = V_{3P}\frac{P_1}{K_{3P}+P_1} - V_{4P}\frac{P_2}{K_{4P}+P_2} - k_3P_2T_2 + k_4C - v_{dP}\frac{P_2}{K_{dP}+P_2} - k_dP_2$$

$$\frac{dM_t}{dt} = v_{sT}\frac{K_{IT}^n}{K_{IT}^n+C_N^n} - v_{mT}\frac{M_T}{K_{mT}+M_T} - k_dM_T$$

$$\frac{dT_0}{dt} = k_{sT}M_T - V_{1T}\frac{T_0}{K_{1T}+T_0} + V_{2T}\frac{T_1}{K_{2T}+T_1} - k_dT_0$$

$$\frac{dT_1}{dt} = V_{1T}\frac{T_0}{K_{1T}+T_0} - V_{2T}\frac{T_1}{K_{2T}+T_1} - V_{3T}\frac{T_1}{K_{3T}+T_1} + V_{4T}\frac{T_2}{K_{4T}+T_2} - k_dT_1$$

$$\frac{dT_2}{dt} = V_{3T}\frac{T_1}{K_{3T}+T_1} - V_{4T}\frac{T_2}{K_{4T}+T_2} - k_3P_2T_2 + k_4C - v_{dT}\frac{T_2}{K_{dT}+T_2} - k_dT_2$$

$$\frac{dC}{dt} = k_3P_2T_2 - k_4C - k_1C + k_2C_N - k_{dC}C$$

$$\frac{dC_N}{dt} = k_1C - k_2C_N - k_{dN}C_N$$

A quantidade total (não conservada) da proteína PER, P_t e da proteína TIM, T_t são dadas por:

$$\begin{aligned} P_t &= P_0 + P_1 + P_2 + C + C_N \\ T_t &= T_0 + T_1 + T_2 + C + C_N \end{aligned} \tag{.1}$$