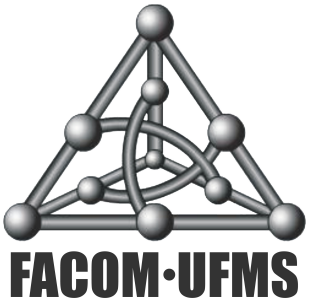
Inferência de redes de regulação gênica a partir de dados de expressão gênica e conhecimento biológico *a priori*

Camila Yumi Koike

Dissertação apresentada
à
Faculdade de Computação
da
Universidade Federal de Mato Grosso do Sul
para
obtenção do título de Mestre
em
Ciência da Computação



Orientador: Prof. Dr. Carlos Henrique Aguena Higa

Área de Concentração: Ciência da Computação

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

Campo Grande, Agosto de 2016

Inferência de redes de regulação gênica a partir de dados de expressão gênica e conhecimento biológico *a priori*

Esta é a versão original da dissertação elaborada pela
Camila Yumi Koike, tal como
submetida à Comissão Julgadora.

Comissão Julgadora:

- Prof. Dr. Carlos Henrique Aguena Higa (orientador) - UFMS
- Pesquisador Dr. Leonardo Rippel Salgado - EMBRAPA
- Prof. Dr. Said Sadique Adi - UFMS

Agradecimentos

Primeiramente, agradeço a Deus e ao universo pela vida e energia para viver.

Agradeço aos meus pais, Tatsuhiro e Cecília, pelo amor, dedicação e por sempre contribuir grandemente na minha formação pessoal e acadêmica.

Ao meu namorado, Leandro, pelo amor, paciência e motivação nos momentos difíceis.

Agradeço aos demais familiares e amigos pelo incentivo.

Agradeço grandemente ao meu orientador, Carlos H. A. Higa, por me orientar e contribuir a minha formação profissional e pessoal. Sempre me orientando com muita dedicação e atenção, respeitando meus limites e compreendendo a vida do outro como um todo. A esse exemplo de ensino, valores, novos entendimentos, conhecimento e formas de pensar. Muito obrigada.

Agradeço ao meu orientador do Trabalho de Conclusão de Curso na graduação, Nalvo F. de Almeida-Jr por ter me incentivado a fazer o mestrado, caso contrário teria continuado a trabalhar nas empresas. E hoje trabalhando na área de docência nas escolas vejo que fiz a escolha certa.

À secretaria de pós-graduação da FACOM-UFMS, por sempre responder prontamente todos os e-mails e com muita eficiência e rapidez entregar os documentos requisitados.

À CAPES pelo apoio financeiro durante o desenvolvimento deste trabalho. E a FACOM-UFMS por toda a estrutura e oportunidade.

Resumo

KOIKE, C. Y. **Inferência de redes de regulação gênica a partir de dados de expressão gênica e conhecimento biológico *a priori***. 2016. 120 f. Dissertação (Mestrado) - Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, 2016.

Neste trabalho, nós estudamos um problema comum na Biologia Sistêmica, o qual é a inferência ou engenharia reversa da rede de regulação gênica a partir de dados de expressão gênica. Nós endereçamos este problema usando o formalismo Booleano, onde a expressão de um gene é representado apenas por dois possíveis valores: 0 (não expressado) ou 1 (expressado). Além disso, nossa metodologia é baseada na abordagem de seleção de característica e utilizamos um algoritmo chamado IFFS - improved forward floating selection. Realizamos experimentos para comparar 2 medidas de interação gênica usada na função critério do algoritmo, o coeficiente de determinação e a informação mútua computada através da entropia de Shannon e de Tsallis. Também incorporamos uma fonte de conhecimento biológico a priori das interações dos genes a partir de um banco de dados chamado STRING. Para validar a metodologia, utilizamos dados das competições DREAM e um conjunto de dados de um estudo do ciclo-celular da levedura. Os resultados mostraram que, geralmente, a informação mútua desempenha um resultado levemente melhor do que o coeficiente de determinação, e que incorporando o conhecimento biológico melhora os resultados. Todo esse conjunto de algoritmos estão implementados na linguagem R e futuram estarão completamente disponível como o pacote Measures_GRN do R em https://github.com/camila-koike/Measure_GRN.

Palavras-chave: Redes de Regulação Gênica, Biologia Computacional.

Abstract

KOIKE, C. Y. **Inference of gene regulatory network from gene expression data and biological knowledge *a priori***. 2016. 120 f. Dissertação (Mestrado) - Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, 2016.

In this work, we studied a common problem in Systems Biology, which is the inference or reverse engineering of gene regulatory networks from gene expression data. We addressed this problem using the Boolean formalism, where the expression of a gene is represented only by two possible values: 0 (not expressed) or 1 (expressed). Besides that, our methodology is based on a feature selection approach and we used an algorithm named IFFS - improved forward floating selection. We performed experiments to compare two measures of gene interactions used in the criterion function of the algorithm, the coefficient of determination and the mutual information computed via Tsallis entropy. Besides that, we also incorporated a biological prior knowledge source of gene interactions from a database known as STRING. To validate the methodology, we used data from the DREAM challenge and a dataset from a budding yeast cell-cycle study. The results showed that, generally, the mutual information performs slightly better than the coefficient of determination, and that incorporating biological knowledge improves the results. All this set of algorithms are implemented in R and hereafter will be available as the Package Measures_GRN for R in https://github.com/camila-koike/Measure_GRN.

Keywords: Gene Regulatory Network, Computational Biology.

Sumário

Lista de Abreviaturas

API	Interface de programação para o usuário (*Application Programming Interface*)
AUPR	Área sob a curva *Precision-Recall*
AUROC	Área sob da curva ROC
CB	Conhecimento biológico
cDNA	DNA complementar
CoD	Coeficiente de determinação
DNA	Ácido desoxirribonucleico (*Deoxyribonucleic Acid*)
DREAM	Organização sem fins lucrativos *Dialogue for Reverse Engineering Assessments and Methods*
FN	Falso negativo
FP	Falso positivo
FPR	Taxa falso positivo
GRN	Rede de regulação gênica (*Gene Regulatory Network*)
IFFS	Algoritmo de seleção de característica (*Improved Forward Floating Selection*)
IM	Informação mútua
KEGG	Banco de dados biológico (*Kyoto Encyclopedia of Genes and Genomes*)
mRNA	RNA mensageiro
NCBI	Banco de dados biológico (*National Center of Biotechnology Information*)
PRE	Precisão
REC	*Recall*
RNA	Ácido ribonucleico (*Ribonucleic Acid*)
rRNA	RNA ribossômico
ROC	Curva de características operacionais do receptor (*Receiver Operating Characteristic*)
SAGE	Tecnologia para quantificação de mRNA (*Serial Analisais of Gene Expression*)
SBS	Algoritmo de seleção de característica (*Sequential Backward Selection*)
SFS	Algoritmo de seleção de característica (*Sequential Forward Selection*)
SFFS	Algoritmo de seleção de característica (*Sequential Forward Floating Selection*)
STRING	Banco de dados biológico (*Search Tool for the Retrieval of Interacting Genes/Proteins*)
TN	Verdadeiro negativo
TP	Verdadeiro positivo
TPR	Taxa verdadeiro positivo
tRNA	RNA transportador

Lista de Símbolos

$\theta(y, Z)$	Coeficiente de Determinação entre o gene y e o conjunto Z.
$\iota(y, Z)$	Informação Mútua entre o gene y e o conjunto Z.
$\sigma(y, Z)$	Conhecimento biológico entre o gene y e o conjunto Z.
$\iota_q(y, Z)$	Informação Mútua baseada na entropia de Tsallis entre o gene y e o conjunto Z.
$\varepsilon_\bullet$	Erro de Bayes cometido ao estimar o valor de y na ausência de observações.
ε_Z	Erro de Bayes ao estimar y dado o conjunto de observações dos genes em Z.
ε'_Z	Erro de Bayes ao estimar y dado o conjunto de observações dos genes em Z penalizando as instâncias não observadas.
ε''_Z	Erro de Bayes ao estimar y dado o conjunto de observações dos genes em Z penalizando as instâncias raramente observadas.
$H(y)$	Entropia de Shannon do gene y.
$H_q(y)$	Entropia de Tsallis do gene y.
$H(y\|Z)$	Entropia condicional de Shannon do gene y dado o conjunto Z.
$H_q(y\|Z)$	Entropia condicional de Tsallis do gene y dado o conjunto Z.
$H'(y\|Z)$	Entropia condicional de Shannon com penalização em instâncias não observadas do gene y dado o conjunto Z.
$H''(y\|Z)$	Entropia condicional de Shannon com penalização em instâncias raramente observadas do gene y dado o conjunto Z.
$H'_q(y\|Z)$	Entropia condicional de Tsallis com penalização em instâncias não observadas do gene y dado o conjunto Z.
$\theta'(y, Z)$	Coeficiente de Determinação com penalização em instâncias não observadas do gene y dado o conjunto Z.
$\theta''(y, Z)$	Coeficiente de Determinação com penalização em instâncias raramente observadas do gene y dado o conjunto Z.
$\iota'(y, Z)$	Informação Mútua com penalização em instâncias não observadas do gene y dado o conjunto Z.
$\iota'_q(y, Z)$	Informação Mútua baseada na entropia de Tsallis com penalização em instâncias não observadas do gene y dado o conjunto Z.
$\iota''(y, Z)$	Informação Mútua baseada na entropia de Shannon com penalização em instâncias raramente observadas do gene y dado o conjunto Z.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Cada célula de um organismo carrega em seu núcleo o mesmo conjunto de genes — segmentos do DNA responsáveis por determinar a síntese de proteínas — . Entretanto, apenas uma fração desses genes são expressos em determinados momentosE conforme o tipo de célula, gerando assim uma diferenciação de expressão dos genes [38].

A expressão gênica é a estimativa da quantidade de RNA mensageiro (mRNA) presente na célula, onde cada molécula de mRNA produzida equivale a um gene transcrito — informação contida em uma sequência específica do DNA. Como cada molécula de mRNA contém a informação de um gene, cada proteína sintetizada a partir dele é produto deste gene. De forma geral, podemos dizer que a expressão de um gene atua de forma funcional em uma célula, já que a partir dele se tem as proteínas, as quais são responsáveis por diversas funções celulares [21].

A diferenciação de expressão gênica é importante porque é ela quem determina as diferentes funções celulares do organismo [17]. Por exemplo, apesar de uma célula do pâncreas possuir o mesmo DNA que uma célula do fígado, ela não necessita sintetizar as mesmas proteínas do fígado justamente porque esses dois tipos de células possuem funções diferentes. Entretanto, existem casos onde a diferenciação da expressão gênica é prejudicial, por exemplo quando uma célula saudável começa a sintetizar proteínas diferentes das quais ela deveria sintetizar, tornando-se uma célula cancerígena.

Para medir o nível de expressão gênica, diferentes técnicas podem ser empregadas, dentre as mais conhecidas estão o SAGE (do inglês *Serial Analisais of Gene Expression*) [59], DNA *microarrays* [42] e o RNA-Seq [61]. Atualmente essas técnicas são as mais conhecidas justamente porque possibilitaram estimar o nível de expressão de vários genes simultaneamente. Estas técnicas são melhores do que as que conseguiam estimar a expressão gênica de apenas um gene por vez, por exemplo a técnica Northern Blots [2].

Conhecendo os genes que estão sendo expressos em diferentes intervalos de tempo é possível gerar hipóteses de como os genes se regulam durante determinado processo ou ciclo celular. Desta forma, modelar e reconstruir uma rede de regulação gênica (GRN) permite conhecer os mecanismos de interação e controle molecular que ocorrem durante determinado ciclo/processo de um organismo. Entretanto, o número de genes (variáveis) é quantitativamente muito maior comparado às amostras (experimentos) disponíveis, sendo isso um grande desafio para a Bioinformática descobrir os inter-relacionamentos entre essas variáveis.

Na tentativa de solucionar esse problema, diversos modelos matemáticos de GRN foram propostos na literatura, entre os mais estudados e aplicados está o modelo de redes Booleanas [27],

redes Bayesianas [22], equações diferenciais ordinárias [26] e redes de associação [11]. Cada modelo, apesar de suas particularidades, ainda deixam a desejar na eficácia de reconstruir GRNs, devido obviamente a grande quantidade de genes para pequenas quantidade de amostras. Apesar dos avanços, o problema de reconstruir GRNs ainda é considerado um problema a ser tratado na área de Biologia Sistêmica e Bioinformática. Não podemos dizer que existe um algoritmo ótimo devido à complexidade do problema. Em geral, a partir de um conjunto de dados de expressão gênica, podem existir centenas ou milhares de redes que sejam consistentes com estes dados. Sendo assim, inferir redes a partir de dados de expressão gênica é um problema inverso mal posto (*ill-posed inverse problem*).

Além dos dados de expressões gênica obtidos através das técnicas citadas acima, existem diversos bancos de dados públicos, como o STRING - *Search Tool for the Retrieval of Interacting Genes/Proteins* [53], NCBI - *National Center of Biotechnology Information* [41], KEGG - *Kyoto Encyclopedia of Genes and Genomes* [25], Gene Mania [62], GenBank [5], entre outros, que reúnem um grande volume de dados sobre os genes. Todos eles contribuem para melhor compreensão do funcionamento e das inter-relações biológicas entre os genes/proteínas. Estes dados, que aqui chamamos de conhecimento biológico *a priori*, podem ser utilizados para auxiliar no processo de inferência de redes.

Assim, uma combinação dos experimentos de expressão gênica juntamente com uma coleta de conhecimento sobre os genes e suas relações, que já foram obtidas por outros pesquisadores, tem surgido recentemente como uma nova abordagem de solução para inferir GRNs de forma mais precisa [18, 24, 35, 50], sendo esse tipo de abordagem o foco de pesquisa deste trabalho. A principal motivação deste trabalho é prover uma metodologia para inferir GRNs, gerando conhecimento e hipóteses de regulação gênica, que podem ser posteriormente verificadas na bancada (*wetlab*).

A inferência de uma GRN pode ajudar a elucidar o funcionamento de um determinado processo biológico dentro da célula, e assim auxiliar no desenvolvimento de novas drogas para o tratamento de doenças. Por exemplo, o câncer não é apenas uma doença causada por genes singulares, mas sim por conjunto deles [19] e redes de interação molecular e de controle [16]. Reconstruir redes de regulação gênica em tecidos saudáveis e doentes é importante para o entendimento dos fenótipos cancerígenos e elaboração de terapias. E considerando a evolução temporal dos níveis de expressão gênica, *i.e.*, sua dinâmica, consegue-se reconstruir um caminho que pode auxiliar em um melhor entendimento dos mecanismos de controle regulatório.

1.1 Objetivos

O primeiro objetivo deste trabalho, foi propor uma metodologia para a inferência de GRN através de um modelo simples porém eficaz, de modo que acrescente conhecimento biológicos aos dados de expressão gênica a fim de gerar melhores resultados. Por esta proposta, implementou-se os algoritmos em uma ferramenta e validou os resultados obtidos. E o segundo objetivo, foi publicar um conjunto de métricas para a inferência de GRN, como um pacote do R, a fim de auxiliar quem deseja comparar o desempenho dessas métricas.

1.2 Contribuições

As principais contribuições deste trabalho são as seguintes:

- Uma metodologia para resolução do problema de reconstrução de GRNs.
- Implementação dos algoritmos de seleção de características e medidas para estimar o nível de predição entre os genes.
- Conjunto dessas funções implementadas em R (publicação de um pacote R).
- Avaliação empírica dos resultados obtidos através de diferentes combinações das medidas implementadas.
- Comparativo entre as medidas comumente utilizadas para inferir GRNs.

1.3 Organização do Trabalho

Incluindo esta introdução, esta dissertação possui ao total 6 capítulos e 1 apêndice. No Capítulo 3 é definido o problema a ser abordado. No Capítulo 2, abordamos os fundamentos teóricos sobre Biologia Molecular e computação necessários para o entendimento do tema abordado. Em seguida, no Capítulo 4, apresentamos a metodologia proposta como tema da dissertação. No Capítulo 5 estão detalhados os principais testes realizados juntamente com os resultados obtidos. Finalmente, no Capítulo 6, discutimos algumas conclusões obtidas neste trabalho através das medidas de inferência. No Apêndice A informamos o relatório com todos os dados completos de cada teste realizado.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão introduzidos alguns conceitos sobre biologia molecular, ciência da computação e notações relacionadas com redes de regulação gênica para melhor compreensão do trabalho como um todo.

2.1 Conceitos de Biologia

Uma célula eucarionte é uma unidade estrutural e funcional básica do ser vivo, com capacidade de replicação. Ela carrega em seu interior o **genoma** do ser vivo, que é basicamente a informação genética que define suas características e funções. Dentro do genoma de cada célula está presente o **DNA**, que por sua vez é formado por **nucleotídeos**, os quais são estruturas formadas por carbonos, fosfato e uma base nitrogenada (adenina, citosina, guanina e timina). Os átomos de carbono formam uma pentose, e para facilitar sua orientação, cada átomo do carbono é numerado de 1' a 5'. Assim, a base nitrogenada liga-se à pentose pelo carbono 1' e ao fosfato pelo carbono 5'. E um nucleotídeo ligá-se a outro por meio de uma ligação entre o carbono 3' de uma pentose a um fosfato o qual se liga ao carbono 5' da pentose seguinte. Quando a cadeia de DNA se inicia com o carbono 3' ligado ao próximo nucleotídeo e termina com o carbono 5' sem ligação a outro nucleotídeo, dizemos que essa cadeia tem orientação 3'→ 5' e vice-versa. A orientação leva em conta se a extremidade possui o carbono 3' ou o 5' sem ligação com outro nucleotídeo, normalmente chamadas de, respectivamente, **extremidade 3' e 5'**. Veja na Figura 2.1 duas fitas de nucleotídeos, cada uma com orientação 3'→ 5' e 5'→ 3'. Repare que essas duas fitas de nucleotídeos estão interligadas por pontes de hidrogênio nas bases nitrogenadas, possuindo cada base sua respectiva base compatível para ligação. Essas duas fitas de nucleotídeos interligadas formam uma estrutura helicoidal chamada de estrutura do DNA [38].

2.1.1 DNA, RNA e Proteína

As relações de informação entre o DNA, o RNA e a proteína são entrelaçadas. O DNA presente no genoma possui a informação necessária para cada um dos produtos gênicos (ex: proteínas) que o organismo pode fazer [17]. Essa informação se encontra codificada nos genes e são estáticas (inserida na sequência do DNA), necessitando de outra molécula intermediária para carregar a informação: o RNA [17]. O RNA é sintetizado a partir de uma cópia de um gene distinto utilizando a sequência do DNA como molde. Esse fluxo de informações é conhecido como o Dogma Central da Biologia,

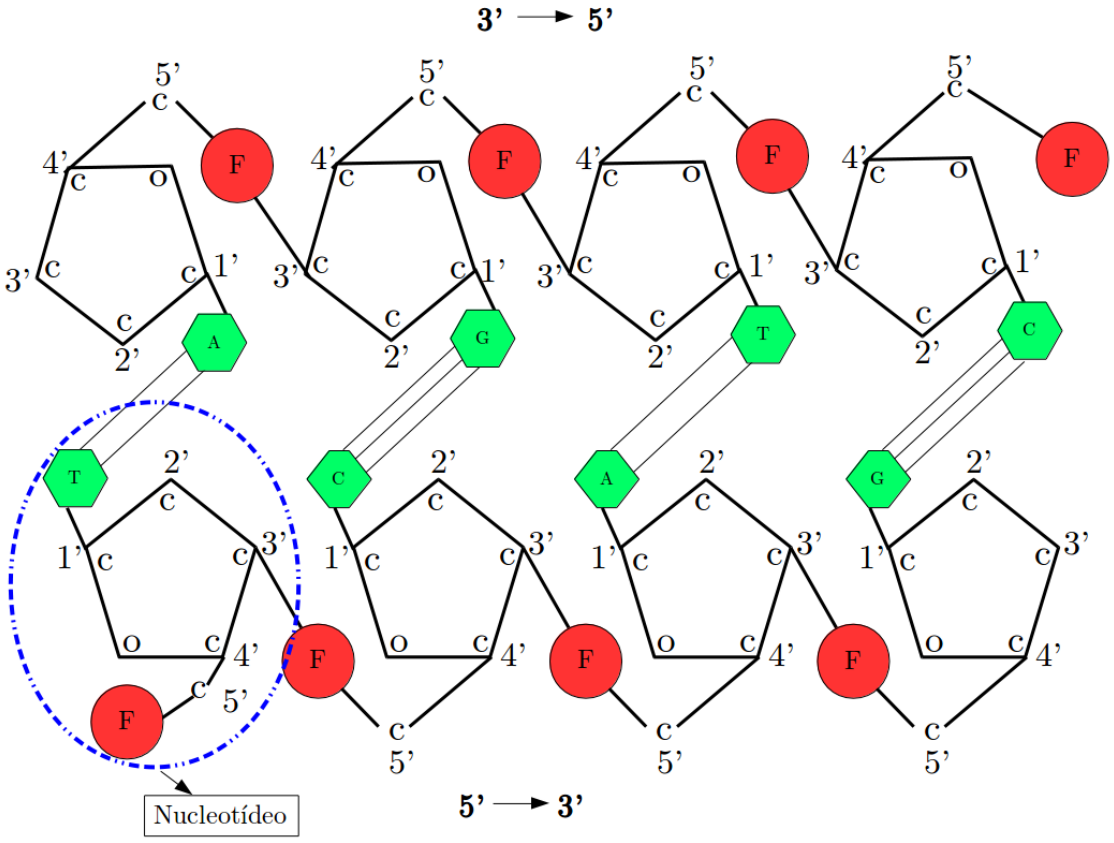


Figura 2.1: *A estrutura do DNA: uma representação bidimensional das duas fitas complementares do DNA, observe que cada fita possui orientação inversas.*

onde a informação pode passar de DNA para DNA (ex: divisão celular) ou de DNA para proteínas, mas não de proteínas para DNA [17].

A informação genética é armazenada no DNA por meio de um código no qual a sequência de bases adjacentes, basicamente, determina a sequência de aminoácidos no polipeptídeo codificado. Primeiro, o RNA é sintetizado a partir do modelo do DNA por um processo conhecido como transcrição. O RNA, carregando a informação codificada sob a forma chamada de RNA mensageiro (mRNA), é então transportado do núcleo para o citoplasma, onde a sequência do RNA é decodificada, ou traduzida, para determinar a sequência de aminoácidos da proteína que está sendo sintetizada [38]. O processo de tradução ocorre nos ribossomos, que são organelas citoplasmáticas com locais de ligação para todas as moléculas de interação, incluindo o mRNA, envolvido na síntese de proteínas. Os ribossomos são constituídos de muitas proteínas estruturais diferentes em associação com tipos especializados de RNA conhecidos como RNAs ribossômicos (rRNA). A tradução envolve ainda um terceiro tipo de RNA, o RNA transportador (tRNA), que fornece a ligação molecular entre o código contido na sequencia de bases de cada mRNA e a sequência de aminoácidos da proteína codificada por tal mRNA [38].

2.1.2 Expressão Gênica

Simplificadamente, um gene pode ser visto como um segmento de uma molécula de DNA que contém um código para uma sequência de aminoácidos de uma cadeia de polipeptídeos e a sequên-

cia reguladora necessária para essa expressão. A maioria dos genes é interrompida por uma ou mais regiões não codificadoras. Essas sequências interpostas, chamadas de **íntrons**, são inicialmente transcritas em RNA (pré-RNA) no núcleo, mas não estão presentes no mRNA final (mRNA maduro) no citoplasma. Então, a informação das sequências intrônicas não é, normalmente, representada no produto proteico final. Os íntrons estão alternados com **éxons**, os segmentos de genes que determinam a sequência de aminoácidos da proteína [38]. O processo de ***splicing*** consiste na retirada dos íntrons de um pré-RNA, de forma a produzir um mRNA maduro, que pode ser mensageiro ou não-codificante [56]. A concatenação de exons, pode ser feita de forma alternativa pela célula, dependendo de um conjunto grande de fatores: do tipo de célula, local, função, etc [1]. No *splicing* alternativo, diferentes éxons de um mesmo pré-RNA podem ser utilizados na produção de diferentes RNAs maduros, e assim gerar proteínas distintas a partir de um único gene, caso essas variações encontrem-se em regiões codificantes [56]. Sua ocorrência permite que informações específicas de um único gene se modifiquem dependendo de sinais do ambiente, gerando transcritos maduros distintos, e conferindo assim uma maior plasticidade à expressão gênica [30, 56].

Existem cinco tipos principais de *splicing* alternativo (ver Figura 2.2): *exon skipping* (uso alternativo de éxon), *mutually exclusive exons* (éxons mutuamente excludentes), *alternative 5' splice sites* (sítios doadores de 5' alternativos), *alternative 3' splice sites* (sítios aceptores de 3' alternativos) e *intron retention* (retenção de íntrons). Além da possibilidade de ocorrer *splicing* alternativo nas seqüências iniciadoras e finalizadoras da transcrição [3].

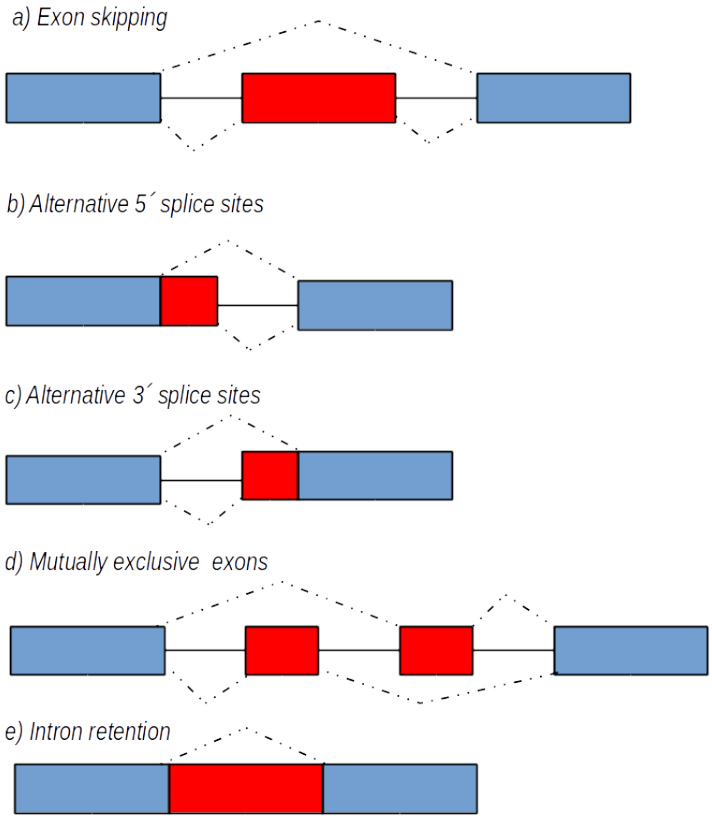


Figura 2.2: *Os cinco tipos principais de splicing alternativo. Éxons constitutivos em azul e alternativos em vermelho, linhas tracejadas representam as possíveis combinações de éxon, Figura modificado de [3].*

O início da transcrição de um gene está sob a influência de promotores (sequências específicas

reconhecidas pela RNA polimerase) e de outros elementos reguladores bem como proteínas específicas conhecidas como **fatores de transcrição**, que interagem com sequências específicas dentro dessas regiões e determinam um padrão espacial e temporal da **expressão de um gene** [38]. Portanto, a expressão gênica neste trabalho é a quantidade de mRNA final no citoplasma, quantidade e diversidade que variam conforme cada tipo de célula e processos em que a célula esteja envolvida. E para se medir essa quantidade de mRNA no citoplasma, existem diversas tecnologias, entre as mais conhecidas temos:

- Northern blot (1977) [2];
- Fluorescent in situ hybridization (1982) [29];
- Serial analysis of gene expression - SAGE (1995) [59];
- DNA *microarray* (1996) [42];
- RNA-Seq (2009) [61];

Como o foco deste trabalho não é explicar o funcionamento de cada tecnologia, vamos explicar somente a do DNA *microarray* porque foi a tecnologia utilizada para extração dos dados de um organismo celular que utilizamos para teste da nossa metodologia.

Para determinar quais genes estão expressos ou não em uma determinada célula através do método de DNA *microarray*, deve-se primeiro recolher as moléculas de mRNA presentes nessa célula em teste e transformá-las em cDNA (DNA complementar) por uma enzima conhecida como transcriptase reversa [42]. Em seguida esses cDNA são ligados a nucleotídeos fluorescentes e postos sobre uma lâmina de microarray de DNA sintético onde cada ponto (*spot*) do array corresponde aos genes conhecidos. Assim os cDNAs marcados que representam mRNAs na célula, vão hibridizar — ou ligar — com seus DNAs sintéticos complementares anexados na lâmina do microarray, deixando seu marcador fluorescente no *spot* [42]. Um processo de lavagem da lâmina remove os cDNAs não hibridizados, e a lâmina passa então por um processo de análise da intensidade do brilho dos *spots*. Se um gene particular é muito ativo, que produz muitas moléculas de mRNA, assim se terá mais cDNAs marcados, os quais hibridizam com o DNA na lâmina de microarray e geram uma área fluorescente muito brilhante [42]. Os genes que são um pouco menos ativos produzem menos mRNA, assim menos cDNA, o que resulta em pontos de menor intensidade da luz fluorescente [64]. Uma ilustração desse processo descrito pode ser vista na Figura 2.3. Os principais fornecedores comerciais de DNA *microarrays* são *Affymetrix, Illumina, Applied Biosystems, GE Healthcare, Beckman Coulter* e *Eppendorf Biochip Systems*.

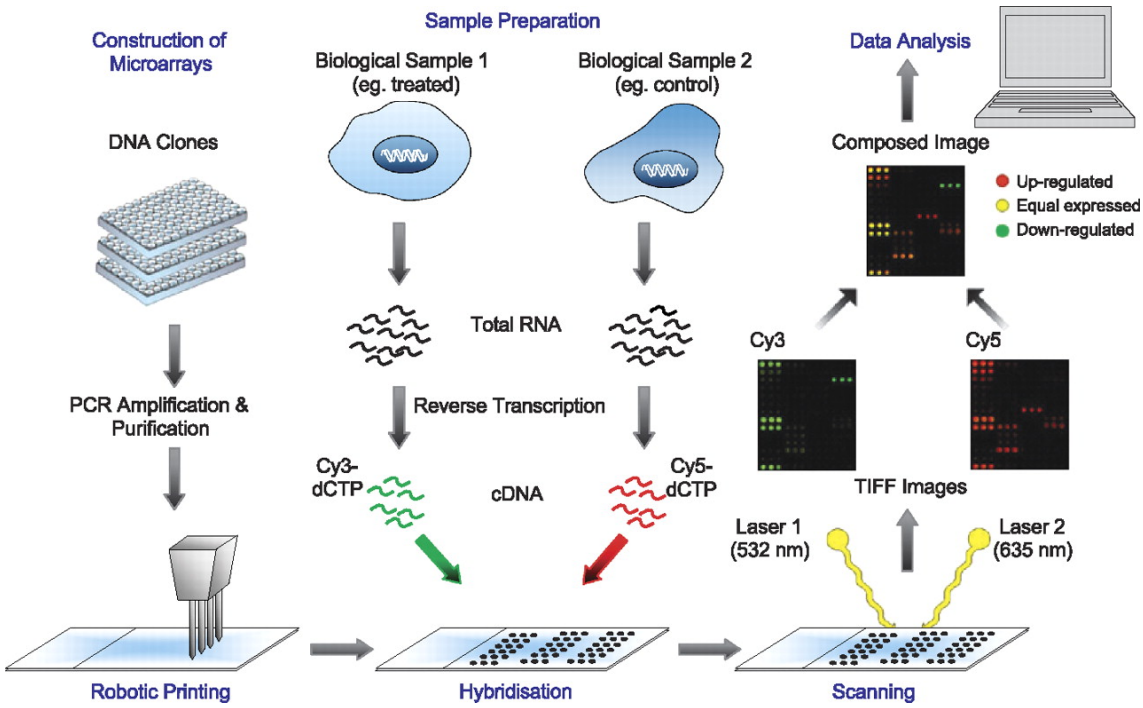


Figura 2.3: *Processo de extração da expressão gênica via microarray. Repare que comparando duas amostras, uma teste de cor verde e a controle de cor vermelha, no processamento de imagem, se a cor vermelha prevalecer significa que o gene da amostra teste foi mais expresso. Caso esteja amarelo o gene do teste se expressou tanto quando o de controle, e se estiver verde o gene de teste não foi expresso tanto quando a de controle. Normalmente, realiza-se o procedimento de microarray com duas amostras (uma de teste e outra de controle) justamente para ver se existem spots de cores diferentes do amarelo, significando assim que já existe diferenciação na expressão gênica. Quando a comparação não interessa, realiza-se o experimento somente com a amostra teste. Figura retirada de [64].*

2.2 Conceitos de Computação

2.2.1 Rede de Regulação Gênica

Uma rede de regulação gênica (GRN) é representado por um diagrama/grafo que representa as interações bioquímicas dos genes/proteínas de maneira simplificada. É uma representação da regulação entre os genes/proteínas envolvidos em determinado processo (ciclo celular, fotossíntese, entre outros) [21]. Essa representação normalmente é feita através de um grafo com nós e arestas, onde os nós representam os genes/proteínas e uma aresta, caso exista, entre um par de nós representa alguma regulação entre eles. As arestas podem ser direcionadas indicando ativação ou inibição de um nó para outro. Uma GRN serve para abstrair as relações existentes entre os genes/proteínas, de forma que se consiga descobrir mais sobre a função de cada gene em um determinado processo químico. A Figura 2.4 ilustra como seria uma GRN a partir dos genes envolvidos.

Em geral, as possibilidades para reconstruir interações regulatórias hipotéticas entre genes são enormes e dependendo da abordagem do problema, existem modelos mais adequados do que outros, pois cada um possui suas particularidades e características. De modo geral, os modelos matemáticos de GRN mais utilizados são: Sistema de equações diferenciais ordinárias [26, 31], Redes Bayesianas [15, 22], Redes Booleanas [28], entre outros. Para este trabalho vamos abordar apenas as Redes Booleanas, para mais informações sobre os diferentes formalismos de modelagem, ver em [11].

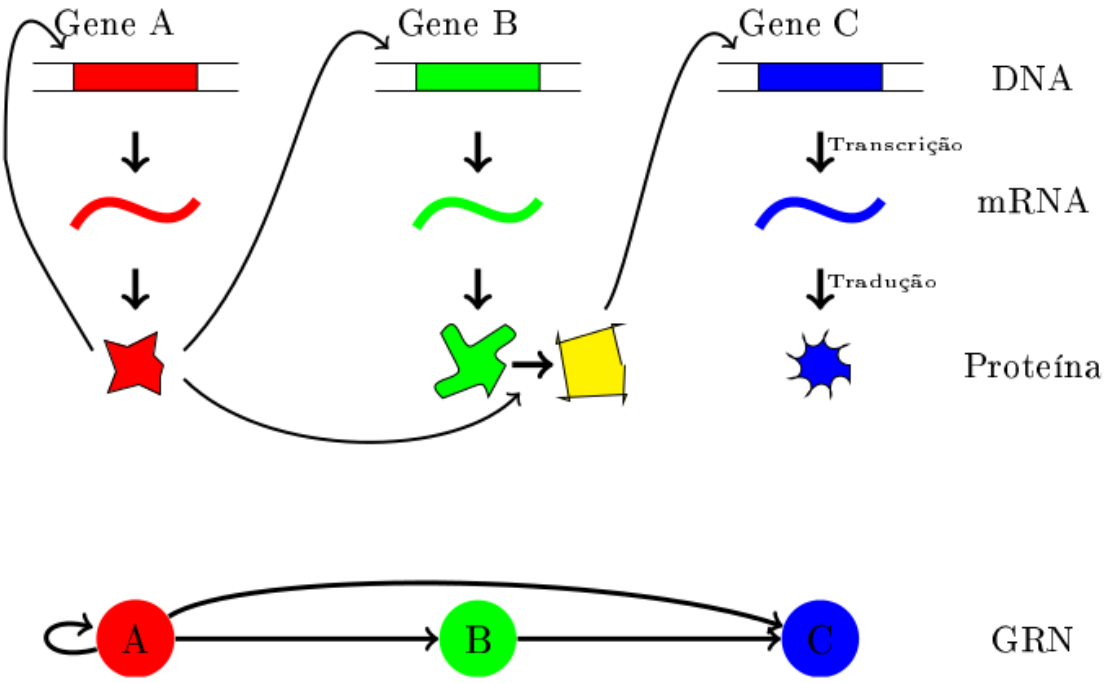


Figura 2.4: *A estrutura de uma GRN. Na parte superior da figura temos uma série de interações bioquímicas ocorrendo entre os genes envolvidos ilustrados em 3 camadas: genoma, transcriptoma e proteoma. Na parte inferior temos a representação dessas interações de forma mais abstrata através de uma GRN. Figura adaptada de [21].*

2.2.2 Redes Booleanas

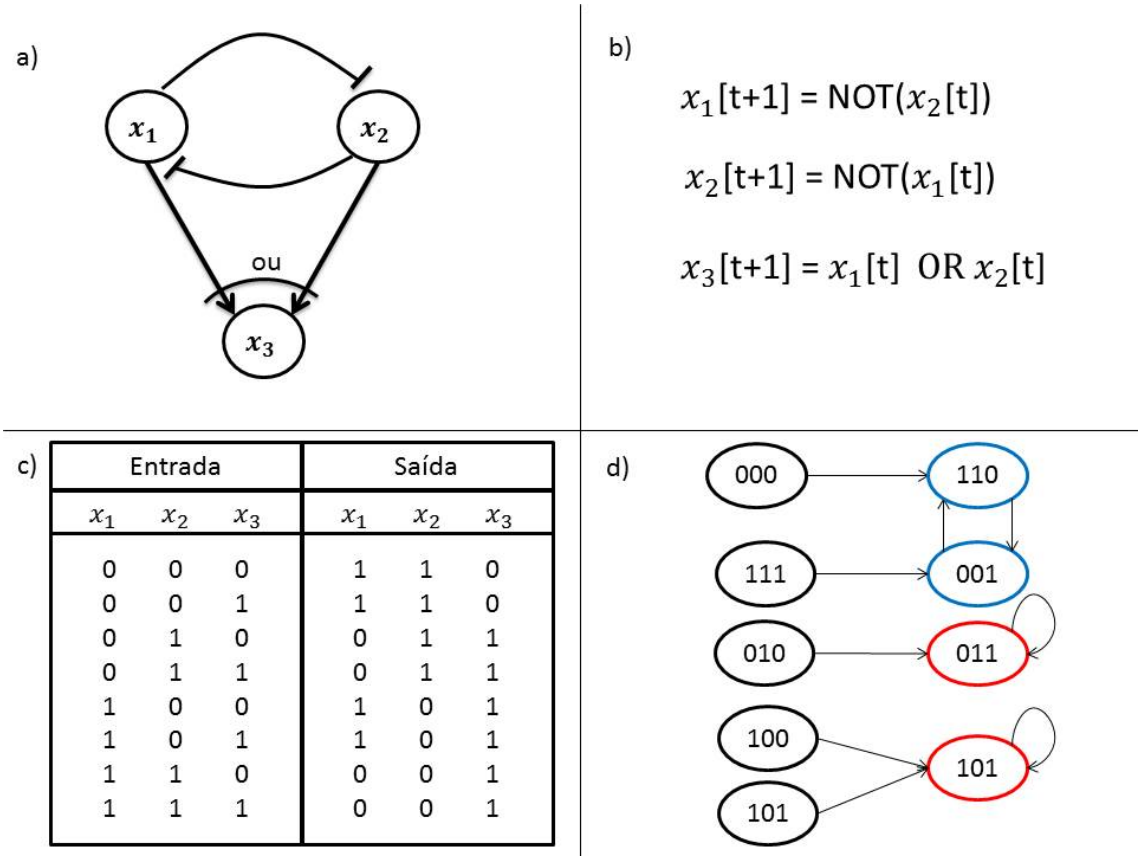
Os modelos discretos destacam-se dos demais porque eles conseguem modelar a rede de regulação gênica de forma mais simples. Apesar da expressão gênica estar em valores contínuos em vez de binário, muitos genes possuem um comportamento biestável (muitos expressos ou não expressos) permitindo o modelo Booleano gerar uma boa aproximação. Por estes motivos, ele será o modelo utilizado neste trabalho.

Uma *rede Booleana* $B(X, F)$ é formada por um conjunto de vértices $X = \{x_1, x_2, \ldots, x_n\}$ e um conjunto de funções Booleanas $F = \{f_1, f_2, \ldots, f_n\}$, onde cada gene x_i, $i = 1, \ldots, n$, pode assumir apenas dois valores: 0 ou 1. O valor do gene x_i no tempo $t + 1$ é determinado por k_i genes, $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ no tempo t através de uma função Booleana $f_i : \{0,1\}^{k_i} \to \{0,1\}$. O mapeamento injetor $j_k : \{1, \ldots, n\} \to \{1, \ldots, n\}, k = 1, \ldots, k_i$ determina a dependência do gene x_i [44]. Dessa forma $x_i(t+1) = f_i(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t))$.

Um *estado* $\mathbf{s}(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ de uma rede Booleana com n genes no instante de tempo t é definido pelos estados correntes (0 ou 1) de todos os n genes no tempo t, ou seja, uma rede Booleana que modela um sistema com n variáveis possui 2^n estados possíveis: $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{2^n-1}$. Dado um estado $\mathbf{s}(t)$, o próximo estado, $\mathbf{s}(t+1)$, é obtido aplicando-se as funções Booleanas em F para cada gene, levando a uma *transição de estados* $\mathbf{s}(t) \to \mathbf{s}(t+1)$. Uma vez que o número de estados é finito e a dinâmica é determinística, após um determinado tempo, certos estados serão revisitados de maneira cíclica. Tais estados formam o que chamamos de *atratores*. Um *atrator* é um ciclo dirigido no diagrama de transição de estados de uma rede. Se o atrator é um estado único, chama-se *singleton*, e se o atrator é composto por mais de um estado é chamado de *ciclo atrator*.

O conjunto de estados que levam a um atrator é chamado *bacia de atração*. Os estados que não conduzem a um atrator são estados visitados apenas uma vez, chamados de *estados transientes*.

Um *diagrama de transição de estados* de rede Booleana com n genes é um grafo dirigido onde os vértices são todos os 2^n estados possíveis da rede e existe uma aresta de $\mathbf{s}_i$ para $\mathbf{s}_j$ se a transição $\mathbf{s}_i \rightarrow \mathbf{s}_j$ ocorre ao aplicarmos as funções em F.



Entrada			Saída		
x_1	x_2	x_3	x_1	x_2	x_3
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	0	1

Figura 2.5: *Exemplo de uma rede Booleana contendo 3 genes. (a) representação na forma de um grafo direcionado, (b) Funções Booleanas, (c) Tabela de transição de estados e (d) Diagrama de transição de estados (estados pretos formam a bacia de atração, de azuis formam um ciclo atrator e de vermelho os singletons)(adaptada de [28]).*

Na Figura 2.5(a), as arestas direcionais com uma seta na extremidade representam uma ativação (x_1 ou x_2 ativam x_3), e as arestas direcionais com uma barra na extremidade representam inibição (x_1 inibe x_2). Na Figura 2.5(b), este mesmo relacionamento entre os genes é expresso em uma equação. A Figura 2.5(c) exibe a representação de todas as possibilidades de estados da rede (entrada, tempo t) e os respectivos resultados (saída, tempo $t+1$). A Figura 2.5(d) exibe o diagrama de transição de estados desta rede, por exemplo, se a rede estiver no estado $(x_1 = 1, x_2 = 0, x_3 = 0)$, então o próximo estado desta rede será $(x_1 = 1, x_2 = 0, x_3 = 1)$.

Neste trabalho, estaremos usando o modelo Booleano no processo de inferência da topologia da rede, mas não estaremos inferindo as funções Booleanas. Em outras palavras, inferimos o grafo como apresentado na Fig. 2.5(a) mas não inferimos as funções apresentadas na Fig. 2.5(b). Apesar disso, é possível adaptar a metodologia para que as funções Booleanas também sejam inferidas.

2.2.3 Algoritmos Seleção de Características

Neste trabalho, o problema de inferência de GRN foi abordado através de um *problema de seleção de características*. Dado um conjunto $X = \{x_1, x_2, \ldots, x_n\}$ de genes, para cada *gene alvo* $y \in X$, desejamos selecionar um subconjunto $Z \subseteq X$ de *genes preditores* (características) que melhor regulam y. Desse ponto de vista, estamos trabalhando com um problema de otimização. Se $x_i \in Z$, temos que a aresta $(x_i, y) \in E$ no grafo $G(X, E)$ que representa a topologia da rede.

Os métodos de seleção de características podem ser separados em dois grupos: os métodos ótimos - métodos de busca exaustiva ou baseado nos princípios de Branch & Bound [46] e viável para problemas de pequeno porte - e os métodos sub-ótimos - métodos que selecionam um bom subconjunto Z com um alto (não necessariamente o maior) valor de uma função critério $J(Z)$. Sem perda de generalidade, vamos assumir que um valor alto de J é melhor do que um valor baixo. Como nos métodos ótimos, testa-se todas as C combinações possíveis do conjunto de característica X, temos que $C_{n,p} = \frac{n!}{p! \cdot (n-p)!}$, para $n = |X|$ e $p = |Z|$, $p = 1, \ldots, |X|$, assim a quantidade de possíveis conjuntos de características que o algoritmo ótimo deve buscar cresce excessivamente quando a dimensionalidade do espaço de características aumenta. Por esta razão, o principal foco deste trabalho são os métodos de buscas sub-ótimos.

Existem diversas técnicas de seleção de características sub-ótima, dentre elas a *Sequential Forward Selection* (SFS) e a *Sequential Backward Selection* (SBS) são as mais utilizadas devido à sua velocidade e simplicidade [37, 55]. O SFS constrói incrementalmente o subconjunto das requeridas características Z iniciando com o conjunto vazio e adicionando uma característica por vez ao subconjunto selecionado de forma que o novo subconjunto maximize J (abordagem *bottom-up*). Já o SBS começa com todas as características X e remove uma por vez do conjunto de características de forma que o subconjunto resultante Z maximize J (abordagem *top-down*). Ambas as abordagens terminam quando o subconjunto selecionado atinge o número desejado de características d [47, 55].

Essas abordagens, apesar de atrativas, sofrem do problema chamado efeito cascata, ou seja, as características descardas da abordagem SBS nunca mais poderão ser selecionadas novamente e as adicionadas pelo SFS não mais poderão ser descartadas da solução parcial.

A primeira tentativa para evitar este problema foi o algoritmo *Plus L take away R* [45] que é basicamente uma combinação de SFS e SBS. Ele acrescenta as características ao subconjunto L vezes, e depois remove as características R vezes até chegar ao tamanho desejado d para o subconjunto de características. Contudo, ele não utiliza nenhuma regra ou condição mútua para selecionar o melhor valor para L e R de forma que obtenha o melhor conjunto de características. Os valores de L e R devem ser determinados pelo usuário. Na versão *bottom-up*, L deve ser maior que R, já na versão *top-down*, $L < R$ [49, 55].

Assim, foi proposto o algoritmo conhecido como Sequential Forward Floating Selection (SFFS) [40] o qual é uma generalização do *Plus L take away R* e que além de evitar o efeito cascata, utiliza uma função critério de relação mútua em que os valores de L e R são determinados e atualizados dinamicamente. O algoritmo SFFS pode ser considerado uma extensão do algoritmo SFS. Em contraste com o SFS, o algoritmo SFFS pode remover características através do SBS, assim conseguindo testar um maior número de combinações de subconjunto de características. É importante ressaltar que a remoção das características incluídas é condicional, tornando-se diferente do algoritmo *Plus L take away R*. A remoção condicional no SFFS só ocorre se o subconjunto de características resultante após a remoção de uma característica particular seja avaliada como "melhor" pela função critério,

ou seja, se sua remoção maximiza J [47, 55].

O SFFS começa a busca com o conjunto de características vazio e usa o algoritmo básico SFS para adicionar uma característica por vez para o subconjunto de características selecionado. Cada vez que uma nova característica é adicionada ao conjunto de características corrente, o algoritmo tenta regredir usando o algoritmo de SBS para remover uma característica de cada vez em busca de encontrar um subconjunto "melhor". O algoritmo para quando o tamanho do conjunto de características corrente é Δ maior do que o número de características d desejado [37, 55]. Isto é necessário para permitir logo em seguida a retirada de características pelo SBS e ainda d ser do tamanho desejado, veja a ilustração do algoritmo na Figura 2.6 .

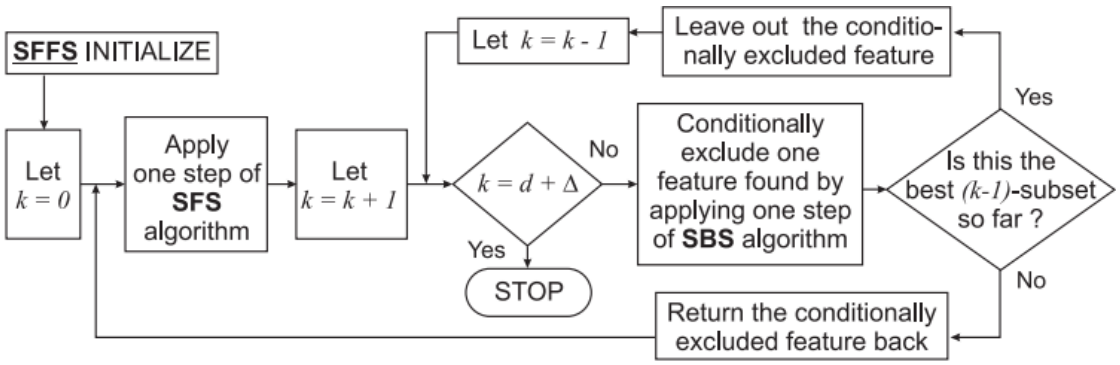


Figura 2.6: *Fluxograma do algoritmo SFFS [47].*

Uma variação do SFFS é o *Improved Forward Floating Selection* (IFFS). Similarmente, este é um algoritmo que seleciona um subconjunto Z de d características de um grande conjunto inicial de X características, onde uma função critério J é usada para comparar a performance do subconjunto de características selecionado.

O IFFS é basicamente o algoritmo SFFS melhorado, porque além dele seguir toda a lógica do SFFS, ele adiciona um passo de busca adicional para verificar se substituindo alguma característica no subconjunto Z corrente por outra características do conjunto X, pode melhorar o conjunto de características corrente, ou seja o valor de J. O IFFS apresenta soluções melhores do que o SFFS por realizar essa busca extra [37]. Pode-se ver os passos ilustrado na Figura 2.7.

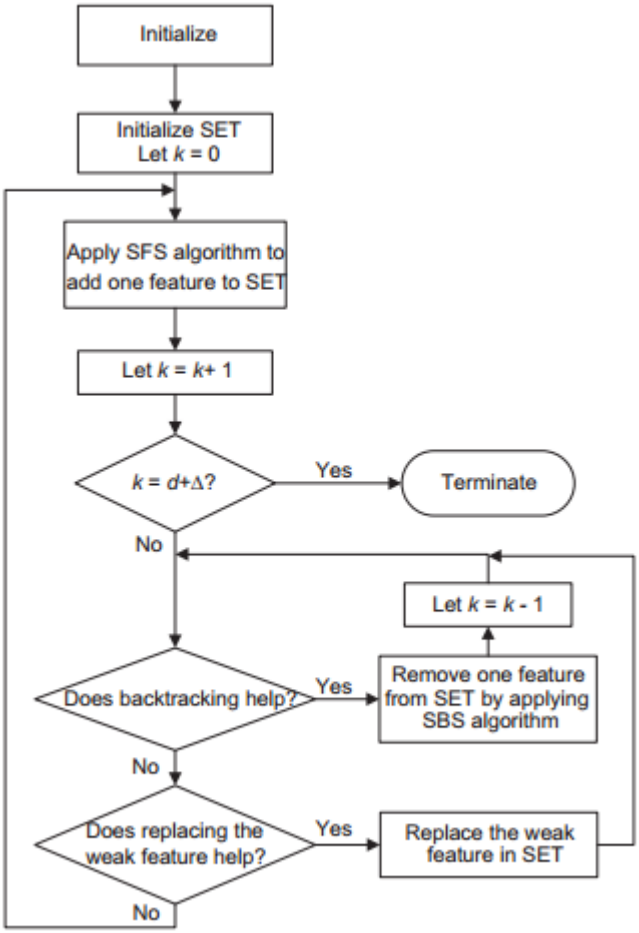


Figura 2.7: *Fluxograma do algoritmo IFFS [37].*

Em relação ao fluxograma apresentado na figura 2.7, nota-se que assim como o SFFS, o IFFS começa a busca com um conjunto vazio e termina quando o conjunto de características corrente atinge $d + \Delta$ características. O método SFS também é usado para adicionar características para o subconjunto Z selecionado. No entanto, o IFFS não só elimina, mas também tenta substituir as características do subconjunto Z selecionado. Para fazer isso, o IFFS verifica se com a remoção de qualquer característica do subconjunto Z e a adição de uma nova (usando o método SFS) para o subconjunto Z, em cada etapa sequencial, melhora o valor J da função critério [37]. Quanto a função critério J, fica a critério do usuário a escolha das medidas que compõem essa função. Na seção seguinte será detalhado as principais medidas utilizadas para formular a função critério J utilizada neste trabalho.

2.2.4 Medidas

Coeficiente de Determinação (CoD)

O coeficiente de determinação (CoD) é uma medida bastante utilizada para selecionar um subconjunto Z de um conjunto de características X que melhor determina o valor de um alvo y.

Definição: *Considere y um gene alvo, $\varepsilon_\bullet$ o erro de Bayes cometido ao estimar o valor de y na ausência de observações e ε_Z o erro de Bayes ao estimar y dado o conjunto de observações dos*

genes em Z. Temos que o CoD de y dado Z é

$$\theta(y, Z) = \frac{\varepsilon_\bullet - \varepsilon_Z}{\varepsilon_\bullet}. \tag{2.1}$$

Quanto maior for o valor de $0 \leq \theta(y, Z) \leq 1$, melhor a predição de y observando Z [12, 13]. Assim deseja-se minimizar o erro ε_Z, pois quanto menor for o erro ε_Z maior será o CoD de $\theta(y, Z)$. Em um cenário real, a distribuição de probabilidade dos genes em X é desconhecida e nós temos que estimar os erros $\varepsilon_\bullet$ e ε_Z a partir dos dados de expressão gênica.

Seja n a quantidade de variáveis binárias aleatórias para predizer outra variável binária qualquer y. Seja T^i a configuração de um vetor qualquer $T = (T_1, T_2, \ldots, T_n)$ cuja expressão binária seja igual a i. Por exemplo, para $n = 2$, $T^0 = 00, T^1 = 01, T^2 = 10$ e $T^3 = 11$; para n variáveis, temos $b = 2^n$ configurações. Assumindo as variáveis aleatórias $x_1, \ldots, x_n$, y possui uma distribuição de probabilidade para cada configuração T^i, de onde calcula-se as probabilidades $r_i = P(T = T^i)$ e $p_i = P(y = 1|T^i)$. Para obter o erro $\varepsilon_\bullet$ considera-se os valores de y sem presença de observações dos genes em Z [20]:

$$\mu = \sum_{i=0}^{b-1} p_i r_i$$

$$\varepsilon_\bullet = \begin{cases} \mu & \text{se } \mu \leq 0.5 \\ 1 - \mu & \text{se } \mu > 0.5 \end{cases}$$

E para calcular o erro ε_Z considera a presença de observações dos possíveis preditores em Z:

$$\varepsilon_Z = \sum_{i=0}^{b-1} min\{p_i, 1 - p_i\} r_i$$

Veja um exemplo na Tabela 2.1 de como calcular os erros $\varepsilon_\bullet$ e ε_Z para $y = x_0$ e $Z = \{x_1, x_2\}$.

Pode-se dizer de uma maneira mais informal que $\varepsilon_\bullet = r_i \cdot p_i$ é o erro de se predizer os valores de y olhando apenas para os valores de y. Por exemplo, na Tabela 2.1, das 23 amostras em 16 delas temos $y = 1$, e em 7 temos $y = 0$, então a chances de y ser igual a 1 é maior, e a chance dessa predição estar errada é de 7/23 que é exatamente o $\varepsilon_\bullet$. E o $\varepsilon_Z = \sum_{i=0}^{b-1} min\{p_i, 1 - p_i\} r_i$ é o erro de predizer os valores de y considerando os estados (configurações) de seus preditores. Por exemplo, na Tabela 2.1, as configurações $T^2 = 10$ e $T^3 = 11$ conseguem dizer com 100% de certeza que o estado do gene alvo será 1, uma vez que $p_i = 1$, o que significa que os genes de Z nestas configurações ativam y (função Booleana) e portanto não há erro nenhum na predição, obtendo-se $\varepsilon_Z = 0$. Entretanto nas configurações $T^0 = 00$ e $T^1 = 01$ já não se tem 100% de certeza de qual será a saída do gene alvo, tendo T^0 e T^1 com respectivamente 46% e 50% de chances de errar, gerando assim um aumento no ε_Z e diminuindo $\theta(y, Z)$.

Penalização de instâncias não observadas e raramente observadas

Um conjunto de dados de expressão gênica é dado por $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$, onde $\mathbf{s}_i \in \{0, 1\}^n$, para $i = 1, 2, \ldots, m$. Como o número de amostras m pode ser insuficiente para uma boa estimação, considera-se que algumas instâncias de Z podem não serem observadas. Assim, penalizamos as instâncias não observadas e raramente observadas.

x_0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	1
x_1	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	0	0
x_2	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0

T^i		r_i	$r_i \cdot p_i$	$r_i \cdot p_i$	$min(p_i, 1-p_i) \cdot r_i$
T^0	00	13/23	7/13	7/23	$6/13 \cdot 13/23 = 6/23$
T^1	01	2/23	1/2	1/23	$1/2 \cdot 2/23 = 1/23$
T^2	10	3/23	3/3	3/23	0
T^3	11	5/23	5/5	5/23	0
				$\mu = 16/23$ $\varepsilon_\bullet = min(\mu, 1-\mu)$	$\varepsilon_Z = 7/23$

Tabela 2.1: *Na tabela superior temos as amostras observadas de $y = x_0$ e $Z = \{x_1$ e $x_2\}$. E na tabela inferior temos o cálculo de $\varepsilon_\bullet$ e ε_Z. Para se calcular a primeira linha considera-se apenas a configuração T^0 onde $x_1 = 0$ e $x_2 = 0$, e conta-se dentro todas as 23 amostras quantas vezes essa configuração apareceu (no caso foram 13), assim temos $r_i = 13/23$. Considerando agora as 13 amostras que estão na configuração T^0 conta-se em quantas delas x_0 foi igual a 1 (no caso foram 7), temos assim $p_i = 7/13$. Posteriormente realiza-se cálculos baseado nos valores de r_i e p_i para encontrar $\varepsilon_\bullet$ e ε_Z. E com esses dois erros calcula-se $\theta(y, Z)$ que neste caso é igual a 0. O que significa que a esse conjunto de possíveis genes preditores $Z = \{x_1$ e $x_2\}$ para o gene alvo $y = x_0$ é bem ruim.*

- **Penalização de instâncias não observadas:** Nem sempre é possível obter instâncias de todas as configurações T^i, assim vale-se a ideia de penalizar as instâncias não observadas. A ideia dessa penalização é incluir as instâncias não observadas no cálculo da função critério recebendo o valor do erro ($\varepsilon_\bullet$) [33]. Assim, uma massa de probabilidade positiva é atribuída às instâncias não observadas e é parametrizada por α. Este parâmetro é adicionado ao número de observações de todas as instâncias possíveis da seguinte fórmula no cálculo de ϵ_Z [33]:

$$\epsilon'_Z = \frac{\alpha(M-N)\epsilon_\bullet}{\alpha M + m} + \frac{\sum_{i=1}^{N}(f_i + \alpha)(1 - \max_{\gamma \in c} P(y = \gamma | z_i))}{\alpha M + m} \tag{2.2}$$

  onde M é o número de todas as possíveis instancias do subconjunto de característica Z, N é o número de instâncias observadas (assim o número de instâncias não observadas é dado por $M - N$), α é o parâmetro de probabilidade que as amostras não observadas vão receber, f_i é a frequência absoluta (número de observações) de z_i e m é o número de amostras.

  Assim sendo, o CoD se torna [33]:

$$\theta'(y, Z) = 1 - \frac{\alpha(M-N)}{\alpha M + m} - \frac{\sum_{i=1}^{N}(f_i + \alpha)(1 - \max_{\gamma \in c} P(y = \gamma | z_i))}{(\alpha M + m)\epsilon_\bullet} \tag{2.3}$$

- **Penalização de instâncias raramente observadas:** Nesta penalização, não é considerado as instâncias não observadas [4]. Ela consiste em modificar a probabilidade condicional de distribuição das instâncias que possui uma única observação. Isso faz sentindo, porque se uma instância z tiver apenas uma observação, o valor de y é totalmente determinado, ou seja, $\theta(y, \mathbf{z}) = 1$, mas a confiabilidade da real distribuição de $P(y = \gamma | Z = \mathbf{z})$ é bem pequena. Assim um parâmetro β atribui um valor de confiança que $y = \gamma$. A principal ideia é distribuir a massa $1 - \beta$ igualmente para todo $P(y \neq \gamma | \mathbf{z})$ e atribuir a massa β para $P(y = \gamma \,|\, \mathbf{z})$ [33].A

probabilidade $P(\mathbf{z}) > \frac{1}{m}$ significa que $\mathbf{z}$ é observado pelo menos uma vez. Adaptando essa penalização temos:

$$\epsilon''_Z = \frac{(M-N)(1-\beta)}{m} + \sum_{\mathbf{z}\in\mathbf{Z}:P(z)>\frac{1}{m}} P(\mathbf{z})(1 - max_{\gamma\in c}P(y=\gamma|\mathbf{z})) \tag{2.4}$$

onde M é o número de todas as possíveis instancias do subconjunto de característica Z, m é o número total de instâncias e N é o número de instâncias $\mathbf{z}$ tais que $P(\mathbf{z}) > \frac{1}{m}$ (mais de uma observação). Assim, $\epsilon_Z = 1-\beta$ quando $P(y|\mathbf{z}) = \frac{1}{m}$, considerando que $\beta \geq 1-\beta$, porque não faz sentindo atribuir massas de probabilidade maiores para as classes não observadas do que para a classe observada.

E o CoD se torna:

$$\theta''(y,\mathbf{z}) = 1 - \frac{(M-N)(1-\beta)}{m\epsilon_\bullet} - \frac{\sum_{\mathbf{z}\in\mathbf{Z}:P(z)>\frac{1}{m}} P(\mathbf{z})(1 - max_{\gamma\in c}P(y=\gamma|\mathbf{z}))}{\epsilon_\bullet} \tag{2.5}$$

Informação Mútua (IM)

A informação mútua é, em geral, um bom critério para mensurar a dependência entre duas variáveis aleatórias x_i e x_j. Essa dependência é quantificada por meio do cálculo da quantidade média de incerteza entre uma variável x_j dado que a ocorrência de outra variável x_i é observada, e vice-versa [65].

Definição: *Considere duas variáveis aleatórias x_i e x_j com distribuição de probabilidade conjunta $p(x_i, x_j)$ e distribuições de probabilidade marginais $p(x_i)$ e $p(x_j)$. A informação mútua $\iota(x_i, x_j)$ é a entropia relativa entre a distribuição conjunta e o produto das marginais $(p(x_i) \cdot p(x_j))$ [9].*

$$\iota(x_i, x_j) = \sum_{x_i,x_j} \left[\; p(x_i, x_j) \cdot log\frac{p(x_i,x_j)}{p(x_i)\cdot p(x_j)} \; \right] \tag{2.6}$$

Para melhor entender a Equação 2.6 de informação mútua, veremos como se utiliza a entropia de Shannon [43] e a entropia de Tsallis [57] para obtê-la.

Entropia de Shannon

Antes de introduzir a entropia de Shannon, abordamos brevemente um pouco da história da entropia. O conceito de entropia foi introduzido por Clausius no contexto da termodinâmica considerando apenas a configuração macroscópica do problema [8]. Posteriormente Boltzmann, no final do século XIX, introduziu o conceito dessa entropia macroscópica em termos de probabilidades associadas à configuração microscopica do sistema [6]. Sua entropia é associada à medida do grau de desordem em sistemas termodinâmicos com a seguinte a expressão [6]:

$$H_B = -k_B \sum_{i=1}^{W} p_i^l n_i \,, \tag{2.7}$$

onde p_i é a probabilidade do sistema ser encontrado no microestado i, k_B é a constante de Boltzmann e W é o número total de microestados acessíveis, dados os vínculos macroscópicos do problema.

Para estados equiprováveis temos $p_i = 1/W$, e assim obtém-se a famosa fórmula $H_B = lnW$ (aqui tomamos a constante de Boltzmann $k_B = 1$).

Utilizaremos o conceito de entropia de informação, que generaliza o conceito de entropia termodinâmica de Boltzmann para uma classe geral de problemas estocásticos. Esse conceito e medida da entropia de informação que envolve medida de desordem, diversidade e incerteza foi introduzido por Claude Elwood Shannon (1948) em uma publicação técnica da empresa de telecomunicações *Bell Telephone Laboratories*, apresentando uma sistematização do conhecimento necessário ao entendimento da eficiência em sistemas de comunicação [43].

A entropia de Shannon equivale a entropia de Boltzmann para $k_B = 1$ e é utilizada para indicar a quantidade de informação contida em uma certa fonte. Considere uma variável qualquer x_i, sua entropia é definida em termos das probabilidades das possíveis ocorrências desta variável aleatória $p(x_i)$, sendo expressa pela seguinte expressão [43]:

$$\begin{aligned} H(x_i) &= \sum_{x_i} (p(x_i) \cdot log\ (1/p(x_i))) \\ &= -\sum_{x_i} (p(x_i) \cdot log\ p(x_i))\,, \end{aligned} \tag{2.8}$$

tal que

$$\sum_{x_i} p(x_i) = 1\,. \tag{2.9}$$

Por exemplo, para um evento binário como do lance de uma moeda (em que a variável aleatória $x_i = moeda$ inclui apenas os eventos cara ou coroa) , temos que o valor de entropia $H(x_i)$ é 1 para uma moeda honesta, veja o cálculo logo abaixo em 2.10:

$$\begin{cases} p_{cara} = p_{coroa} = 0.5 \\ log_2(1/p_{cara}) = log_2(1/p_{coroa}) = 1 \end{cases}$$

$$\begin{aligned} H(x_i) &= p_{cara} * log_2(1/p_{cara}) + p_{coroa} * log_2(1/p_{coroa}) \\ &= 0.5 * 1 + 0.5 * 1 = 1. \end{aligned} \tag{2.10}$$

Repare que a entropia é uma função de distribuição da variável aleatória e não depende dos valores dos estados assumidos por ela, referindo-se apenas às suas probabilidades [9].

A Figura 2.8 mostra a relação entre $p(x_i)$ e $H(x_i)$ em função dos valores de um evento binário (0 ou 1). A entropia é $H(x_i) = 0$ (entropia mínima) quando $p(x_i) = 0$ ou $p(x_i) = 1$, pois nessa distribuição de probabilidade não há incerteza; por um outro lado quando $p(x_i) = 0.5$ (eventos equiprováveis), a incerteza é máxima e a entropia também ($H(x_i) = 1$).

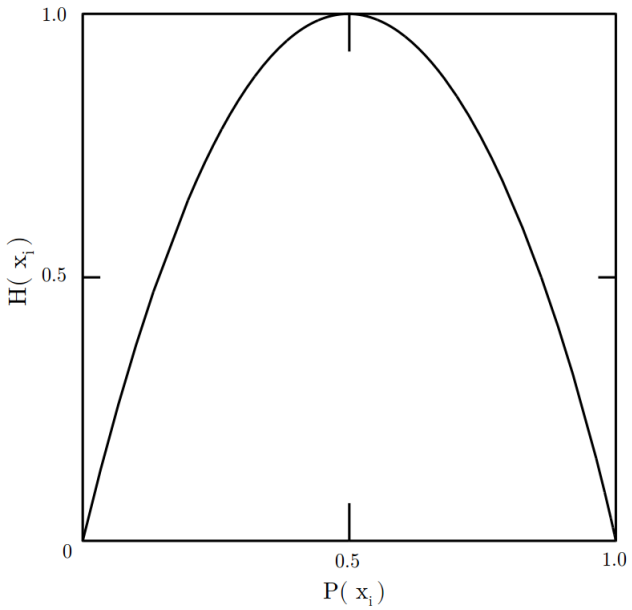


Figura 2.8: *Gráfico da entropia de $H(x_i)$ em função das probabilidade $p(x_i)$ de um evento binário [9].*

A entropia é uma medida de incerteza atrelada a uma variável, ou seja, quanto maior for a entropia de uma variável, maior será a incerteza em predizer o valor dessa variável. Similarmente, a entropia conjunta entre duas variáveis é [9]:

$$H(x_i, x_j) = -\sum_{x_i, x_j} [p(x_i, x_j) \cdot log\ p(x_i, x_j)] \tag{2.11}$$

onde $p(x_i, x_j)$ representa a distribuição de probabilidade conjunta das variáveis aleatórias x_i e x_j.

E a entropia condicional média entre duas variáveis é [9]:

$$H(x_i|x_j) = -\sum_{x_i} \sum_{x_j} [p(x_i, x_j) \cdot log\ p(x_j|x_i)] \tag{2.12}$$

Dadas duas variáveis aleatórias x_i e x_j, a relação entre a entropia H e a informação mútua ι é dada por [9]:

$$\iota(x_i, x_j) = H(x_i) + H(x_j) - H(x_i, x_j)\,, \tag{2.13}$$

$$\iota(x_i, x_j) = H(x_j) - H(x_j|x_i)\,, \tag{2.14}$$

$$\iota(x_i, x_j) = H(x_i) - H(x_i|x_j)\,, \tag{2.15}$$

$$\iota(x_i, x_i) = H(x_i)\,\text{e} \tag{2.16}$$

$$\iota(x_i, x_j) = \iota(x_j, x_i)\,. \tag{2.17}$$

Veja a Figura 2.9 que ilustra a relação descrita nas Equações 2.13 a 2.17.

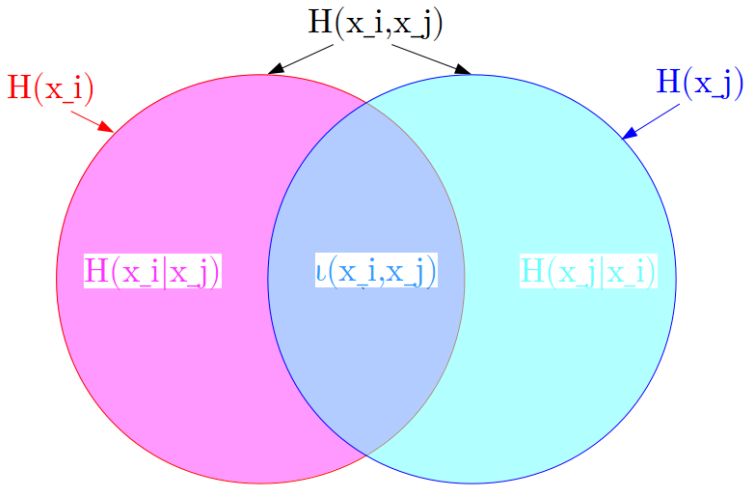


Figura 2.9: *Relação de entropia e informação mútua. Figura adaptada de [9].*

A informação mútua é uma medida da quantidade de informação que uma variável aleatória contém acerca da outra. Por exemplo: Considere duas variáveis aleatórias independentes x_i e x_j com distribuições de probabilidade $p(x_i)$ e $p(x_j)$, temos que $p(x_i, x_j) = p(x_i)p(x_j)$ por elas serem independentes, e portanto $\iota(x_i, x_j) = 0$. Informalmente, podemos dizer que não há informação mútua entre duas variáveis independentes, ou seja não há intersecção entre as entropias de cada variável, assim uma variável em nada ajuda na predição do valor da outra variável possuindo $\iota(x_i, x_j) = 0$.

A informação mútua não se limita apenas aos pares de variáveis. Desse modo, se considerarmos um gene alvo $y \in X$ e um subconjunto de genes $Z \subseteq X$, nós podemos computar $\iota(y, Z) = H(y) - H(y \mid Z)$. O objetivo é encontrar um subconjunto Z que maximize $\iota(y, Z)$, o qual é um subconjunto Z que minimiza a entropia $H(y \mid Z)$. E como no caso do CoD, a entropia também é computada a partir da distribuição de probabilidade estimada a partir dos dados de expressão gênica. Apenas reescrevendo a Equação 2.12, temos que:

$$H(y|Z) = \sum_{\mathbf{z}} p(\mathbf{z}) \cdot H(y|\mathbf{z}) \,, \tag{2.18}$$

onde

$$H(y|\mathbf{z}) = \sum_{\gamma \in c} P(y = \gamma|\mathbf{z}) log\ p(y = \gamma|\mathbf{z}) \,, \tag{2.19}$$

onde c são as de classes de y e $\mathbf{z}$ é uma configuração do vetor formado pelos genes em Z; no caso do modelo Booleano temos $c = \{0, 1\}$.

Entropia de Tsallis

A entropia de Tsallis é uma generalização da Entropia de Boltzmann [34, 57]. Boltzmann introduziu a entropia como uma ponte entre a descrição microscópica e a descrição macroscópica do problema. Entretanto, em sua demonstração matemática existiam algumas condições a respeito da

natureza física do sistema, que necessitavam serem satisfeitas para assegurar as propriedades da entropia macroscópicas de Clausius em sua abordagem microscopica [6]. Por exemplo, para assegurar essa extensibilidade (macroscópio para microscópio) era permitido apenas interações de curto intervalo [6, 58]. Assim, apesar da grande importância e sucesso da entropia de Boltzmann, ainda havia situações em que tais condições não eram preservadas e dificilmente a entropia de Boltzmann conseguiria recuperar a entropia de Clausius. Em outras palavras, podemos dizer que a entropia é uma quantidade extensiva. No entanto, essa hipótese de extensividade não é verificada para todos os tipos de sistemas na prática. Boltzmann descreve os sistemas de mecânica estatística como espaços igualmente provável em qualquer estado possível [6]. Levando-se em conta a possibilidade de que, devido à não-homogeneidade do sistema, como por exemplo, estruturas porosas ou fractais, alguns subespaços sejam preferenciais, C. Tsallis propôs a seguinte generalização [34, 58]:

$$H'_q = -k\frac{1-\sum_i^w p_i^q}{1-q}\,, \tag{2.20}$$

onde k é uma constante positiva, w é o número das distintas configurações do sistema, p_i é a probabilidade de tal configuração e $q \in \mathbb{R}$ é o parâmetro entrópico que caracteriza a extensividade do sistema. Quando $q > 1$ (sub-extensividade), esta expressão dá peso maior às regiões mais prováveis e peso menor às regiões menos prováveis para o cálculo da entropia. Quando $q < 1$ (super-extensividade), o oposto ocorre. E quando $q = 1$ temos a extensividade equivalendo à entropia de Boltzmann e Shannon [58]. Assim, se $q > 0$ então H_q é sempre não-negativo. É possível ter $H_q > 0$ para $q < 0$ mas não é sempre que acontece isso, portanto vamos usar $q > 0$.

Definindo $\ln_q(x) = (x^{1-q} - 1)/(1-q)$, podemos reescrever a Equação 2.20 de forma similar à entropia de Shannon: $H_q = -k\sum_i^w p_i^q \ln_q p_i$. Dessa forma, a *informação mútua generalizada* entre x_i e x_j pode ser definida por [7, 34]:

$$\iota'_q(x_i, x_j) = \sum_{x_i}\sum_{x_j} P(x_i, x_j)\ln_q\left(\frac{P(x_i, x_j)}{P(x_i)P(x_j)}\right). \tag{2.21}$$

Esta *informação mútua generalizada*, assim como na equação 2.15, é maximizada pela minimização de $H_q(x_i \,|\, x_j)$. Para computar a entropia de $H_q(y \,|\, Z)$, as probabilidades devem ser estimadas a partir dos dados.

Penalização de instâncias não observadas e raramente observadas

Dado que algumas configurações em Z pode não estar presente nos dados, é possível penalizar as instâncias não observadas e também aquelas pouco observadas.

- **Penalização de instâncias não observadas:** pela mesma ideia de penalização descrita em 2.2.4, temos que no cálculo da entropia condicional (Equação 2.12), as instâncias não observadas recebem a entropia da distribuição marginal de y ($H(y)$). Considerando esta penalização, a entropia condicional de Shannon é definida por [33]:

$$H'(y|Z) = \frac{\alpha(M-N)H(y)}{\alpha M + m} + \frac{\sum_{i=1}^N (f_i + \alpha)H(y|Z = z_i)}{\alpha M + m}\,, \tag{2.22}$$

onde $\alpha \geq 0$ é o peso da penalização, M é o número de possíveis instâncias do subconjunto de

característica Z, N é número de instâncias observadas, f_i é a frequência absoluta (número de observações) de z_i e m é o número de amostras.

Similarmente, a entropia de Tsallis [34] é definida por:

$$H'_q(y|Z) = \sum_{Z=1}^{N} \frac{r_Z + \alpha}{\alpha M + m} \frac{1 - \sum_y P(y \mid Z)^q}{q - 1}, \tag{2.23}$$

onde r_Z é o número de cada instância observada de Z.

Logo, a IM baseada na entropia de Shannon e Tsallis se tornam, respectivamente [33, 34]:

$$\iota'(y, Z) = H(y) - H'(y|Z) \text{ e} \tag{2.24}$$

$$\iota'_q(y, Z) = H(y) - H'_q(y|Z). \tag{2.25}$$

- **Penalização de instâncias raramente observadas:** seguindo a mesma ideia apresentada em 2.2.4, temos que no cálculo da entropia condicional média $H(y|Z)$ (Equação 2.12), o cálculo de $H(y|\mathbf{z})$(Equação 2.19 se torna[33]:

$$H''(Y|\mathbf{Z} = \mathbf{z}) = \frac{(M - N)}{m} H((\Delta(0), \ldots, \Delta(|c| - 1))) + \sum_{z \in Z : P(z) > \frac{1}{m}} P(z) H(y|\mathbf{z}) \tag{2.26}$$

onde $\Delta : 0, 1, \ldots, |c| - 1 \to [0, 1]$ é a distribuição de probabilidade dado por:

$$\Delta(i) = \begin{cases} \beta & \text{para } i = y, \\ \frac{1-\beta}{|c|-1} & \text{para } i \neq y \end{cases} \tag{2.27}$$

onde c são as de classes de y (e $|c|$ é a cardinalidade de y), N é o número de instâncias $\mathbf{z}$ tais que $P(\mathbf{z}) > \frac{1}{m}$ (mais de uma observação), M é o número de todas as possíveis instancias do subconjunto de característica Z e m é o número total de instâncias. Assim, a IM com penalização de instâncias raramente observadas baseada na entropia de Shannon, fica como [33]:

$$\iota''(y, Z) = H(y) - H''(y|Z) \text{ e} \tag{2.28}$$

Capítulo 3

O problema da Inferência de GRN

Neste capítulo, trataremos de um problema chave na área de Biologia Sistêmica, que é a inferência de redes de regulação gênica a partir de dados de expressão gênica.

3.1 Definindo o problema

Uma rede de regulação gênica é um conjunto de genes e suas interações, que pode ser representada por um modelo matemático que nos auxilie no processo de inferência. Neste trabalho, o modelo utilizado é baseado no formalismo Booleano, onde cada gene pode assumir dois valores possíveis: 0 (quando o gene não está sendo expresso) e 1 (quando o gene está expresso). Além disso, podemos modelar as interações gênicas a partir de um grafo direcionado. Dessa maneira, dado um conjunto $X = \{x_1, x_2, \ldots, x_n\}$ de genes, podemos representar as interações entre os genes de X por um grafo direcionado $G(X, E)$, onde X são os vértices do grafo e E é o conjunto de arestas de tal forma que $(x_i, x_j) \in E$ se o gene x_i participa do processo de regulação do gene x_j.

De maneira informal, o problema de inferência de GRN consiste em descobrir as relações de interação gênica em um grafo, ou seja, as arestas em E, a partir de dados de expressão gênica. Este é um problema inverso mal posto (*ill-posed inverse problem*) visto que podem existir vários grafos que representem os dados tomados como entrada do problema.

Formalmente, o problema é definido da seguinte maneira:

Problema da Inferência de GRN a partir de dados temporais de expressão gênica: *dado um conjunto* $X = \{x_1, x_2, \ldots, x_n\}$ *de genes (vértices), onde* $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$ *e um conjunto de dados de expressão gênica* $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$, *onde* $\mathbf{s}_t \in \{0, 1\}^n$, $t = 1, 2, \ldots, m$, *encontrar as arestas* E *do grafo direcionado* $G(X, E)$ *que represente os dados em* S.

Quando dizemos que um grafo G representa os dados em S, estamos dizendo isso no sentido probabilístico. Existe um conjunto de interações gênicas desconhecidas que governam as expressões gênicas que observamos em S. Como a distribuição de probabilidade das variáveis em X é desconhecida, estimamos essa probabilidade a partir de S e inferimos as interações.

Como estamos considerando um modelo Booleano de GRN, o conjunto de dados de entrada S pode ser visto como uma matriz $n \times m$, onde as linhas correspondem aos genes e cada coluna corresponde a uma amostra em um certo instante de tempo t, ou seja, $s_t = (x_1(t), x_2(t), \ldots, x_n(t))^T$. Quando existe uma relação temporal entre as amostras dizemos que os dados são *temporais*. Por

exemplo, podemos obter as amostras de expressão gênica ao longo do ciclo celular de um organismo; assim, existe uma relação de tempo entre as amostras: $\mathbf{s}_1 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{s}_m$. De outra maneira, pode ser que esta relação de tempo não exista, e neste caso dizemos que os dados são *estacionários*. Por exemplo, se cada amostra $\mathbf{s}_t$ for obtida através do perfil gênico de uma célula cancerígena de organismos distintos.

Capítulo 4

Metodologia

4.1 Dados de entrada e saída

Nesta seção descrevemos os dados de entrada e os dados de saída da nossa metodologia.

4.1.1 Dados de entrada

São dois tipos de dados que esta metodologia processa, sendo o primeiro obrigatório e o segundo opcional:

1. **Expressão Gênica:** a expressão gênica que essa metodologia processa precisa estar quantificada em forma numérica. Sendo assim, os dados de expressão gênica podem ser provenientes de *microarray*, RNA-Seq ou qualquer outra tecnologia, desde que seja possível quantificá-las de forma numérica.

   Para este trabalho foram utilizados dados de expressão gênica extraídos através de DNA *microarray* pois já vem em forma numérica e não requer demasiado pré processamento dos dados como necessitaria no caso do RNA-Seq. O resultado do escaneamento da intensidade da luz fluorescente (DNA *microarray*) são dados contínuos. Assim, para o caso deste trabalho, que recebe como entrada dados numéricos (contínuos) e utiliza um modelo Booleano, realiza-se uma discretização dos dados de expressão gênica conforme explicado na Seção 4.3. Considere, portanto, $X = \{x_1, x_2, \ldots, x_n\}$ o conjunto de genes, onde $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$ e $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ o conjunto de dados de expressão gênica, onde $\mathbf{s}_t \in \{0, 1\}^n$, $t = 1, 2, \ldots, m$, como parte dos dados de entrada para nosso algoritmo.

2. **Conhecimento Biológico (CB):** para incorporar o **conhecimento biológico *a priori*** em nossa metodologia, utilizamos o banco de dados STRING (Search Tool for the Retrieval of Interacting Genes/Proteins). Dentre os bancos de dados analisados, este foi o escolhido porque ele é um banco de dados curado, aqueles bancos de dados mantidos com alto grau de confiabilidade e atualização, e também possui uma API (*Application Programming Interface*) para consulta diretamente ao banco de dados STRING através do pacote em R, chamado STRING DB [53]. Este pacote em R foi muito conveniente para nossa metologia porque ela foi desenvolvida na linguagem R. A linguagem R [23], criada originalmente por Ross Ihaka e por Robert Gentleman em 1996, disponibiliza uma ampla variedade de técnicas estatísticas e gráficas, incluindo modelagem linear e não linear, análise de séries temporais, classificação,

agrupamento, sendo também facilmente extensível através de pacotes. O *Rstudio* [51] foi utilizado como a interface gráfica do utilizador (GUI), para implementar os algoritmos em R.

A versão corrente do STRING é a v10 e ela abrange atualmente mais de 9.000.000 proteínas de mais de 2.000 organismos [52]. A Figura 4.1 mostra as interações entre determinados genes da levedura. As interações incluem associações diretas (físicas) e indiretas (funcionais) que são derivados a partir de quatro fontes: contexto genômico, experimentos de alto rendimento, expressão gênica e conhecimento prévio [54]. O STRING integra quantitativamente dados de interação a partir destas fontes para um grande número de organismos, e as transferências de informação entre esses organismos caso sejam aplicáveis.

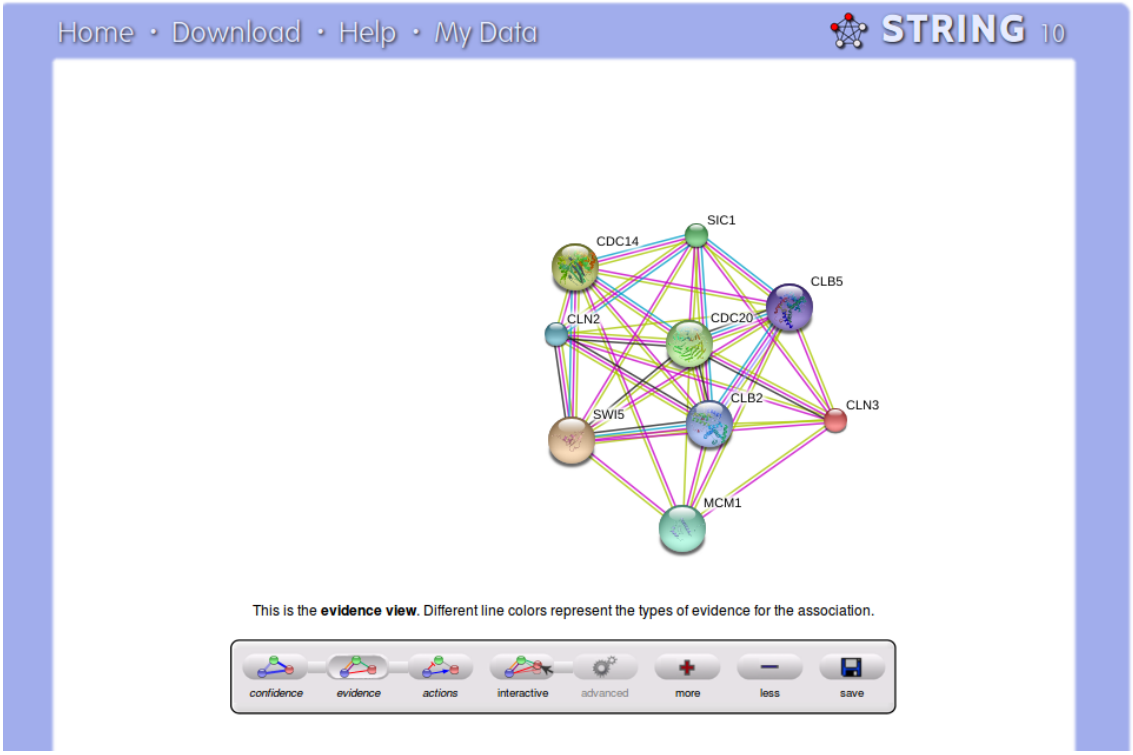


Figura 4.1: *Grafo de interações entre alguns genes do organismo Saccharomyces cerevisiae*

Para cada par de genes de um dado organismo, o banco de dados STRING provê, um conjunto de *score* (pontuação) baseada em: vizinhança conservada, fusão dos genes, co-ocorrência filogenética, co-expressão, experimentos e mineração de texto da literatura. Além desses *scores*, o STRING fornece um *combined score* (pontuação combinada) expressando a confiança quando uma associação é afirmada por diversos tipos de evidências. Para mais detalhes ver na referência [60].

Seja $c(x_i, x_j)$ o *combined score* para um par de genes obtidos do STRING. Primeiro, nós normalizamos os *scores* (ver Seção 4.3.2) para cada par (x_i, x_j). E então, definimos o **Conhecimento Biológico (CB)** para um gene alvo $y \in X$ e um subconjunto de genes $Z \subseteq X$ como:

$$\sigma(y, Z) = \frac{1}{|Z|} \sum_{x_j \in Z} c(y, x_j)\,. \tag{4.1}$$

O *score* fornecido pelo STRING (quando disponível para o organismo em estudo), juntamente

com o CoD e a IM, compõe a função critério para selecionar o melhor subconjunto de genes preditores para um dado gene alvo.

4.2 Dados de Saída

A saída do algoritmo de seleção de características consiste de uma lista L, onde para cada gene em X, são listados d genes preditores (conjunto Z), e seus respectivos *scores* ($J(Z)$). Em outras palavras, L possui n linhas e $d+1$ colunas.

Entretanto, devemos executar o algoritmo de seleção de características para $d = 1, 2, \ldots, k$, lembrando que $d = |Z|$. O parâmetro k limita o grau de entrada dos vértices no grafo que representa a GRN inferida. Assumimos que $k \ll n$ é um número pequeno pois, biologicamente, um gene/proteína não depende de uma grande quantidade de outros genes/proteínas para executar uma determinada função. Além disso, é computacionalmente inviável considerar um valor alto para o parâmetro k. Assim, obtém-se k listas, $L_1, L_2, \ldots, L_k$, para diferentes valores de d, que passam por um pós-processamento (ver Seção 4.5) para resultar em uma única lista U contendo pares de genes (x_i, x_j) (arestas direcionadas) ordenados pelo valor do *score*. Opcionalmente, pode-se gerar o grafo a partir da lista U, obtendo uma visualização como mostra a Figura 4.2.

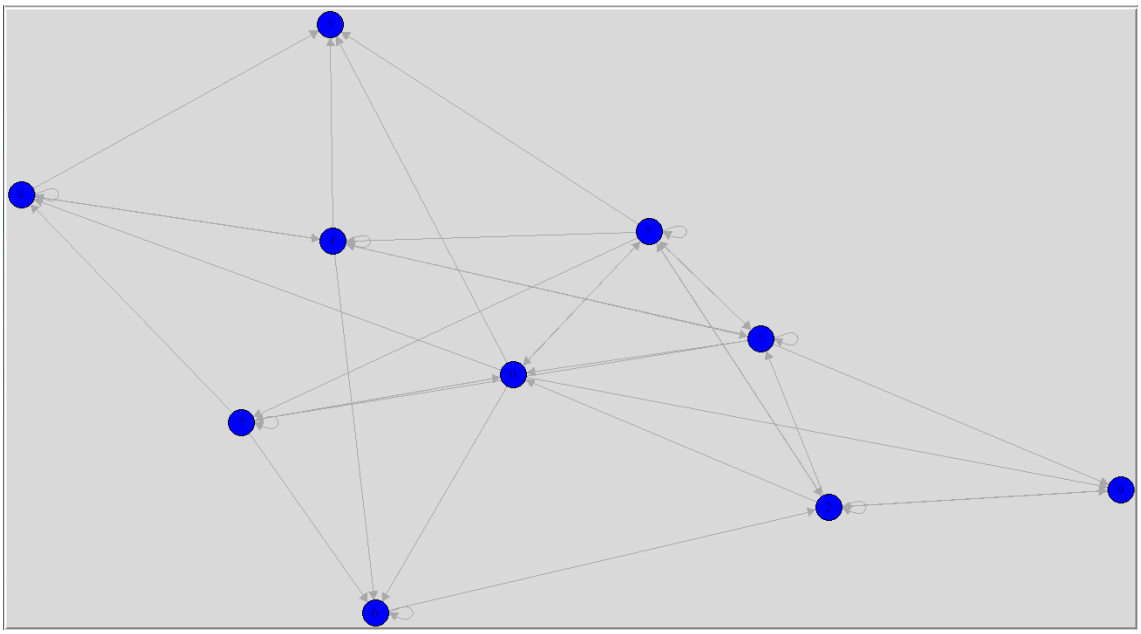


Figura 4.2: *Exemplo de saída: um grafo representado uma GRN qualquer.*

4.3 Pré-processamento dos dados

Como já foi citado na Seção 4.1, os dados de entrada passam por um pré-processamento para então serem processados pelo algoritmo de inferência. Os pré-processamentos utilizados foram:

- Discretização dos dados de expressão gênica utilizando o algoritmo BiKmeans;
- Normalização dos *scores* para cada par (x_i, x_j) fornecido pelo banco de dados STRING.

		Kmeans		
		1	2	3
CoKmeans	1	1	2	3
	2	2	4	6
	3	3	6	9

Tabela 4.1: *Os possíveis produtos de Kmeans e CoKmeans, para $k+1=3$. Produtos de 1 a 3 são colocados no primeiro intervalo e 4 a 9 no segundo.*

4.3.1 Discretização

O algoritmo BiKmeans implementa tanto o algoritmo Kmeans quanto o CoKmeans. Seja $S[n,m]$ a matriz dos dados de séries temporais, onde n é o número de genes e m é o número de amostras de pontos temporais com valores contínuos. $S[n,m]$ corresponde ao valor de expressão do gene n no tempo m, $S[n,*]$ denota ao dado de expressão de todos os instantes de tempo do gene n e $S[*,m]$ denota a expressão de todos os genes no instante de tempo m [32].

- **Kmeans:** divide $S[n,*]$ em k intervalos, agrupando os valores próximos das expressões do gene n no mesmo intervalo [36]. O parâmetro k fica a critério do usuário. Neste trabalho utilizamos $k=2$ pois estamos considerando um modelo Booleano de GRN.
- **CoKmeans:** divide $S[*,m]$ em k intervalos, agrupando os valores próximos das expressões no tempo m no mesmo intervalo [32]. Pelo mesmo motivo do Kmeans, $k=2$.
- **BiKmeans:** ambos, Kmeans e CoKmeans são executados com parâmetro $k+1$, resultando em dois valores discretos para cada expressão gênica. Se o produto dos dois valores for igual ou maior do que x^2 e menor do que $(x+1)^2$, o valor discreto final desta expressão é x, onde x é um inteiro positivo no intervalo 1 a k [32] (ver Algoritmo 1). Por exemplo, se $k=2$ os possíveis valores resultantes do Kmeans e do CoKmeans quando executados com o parâmetro $k+1=3$ são 1, 2 e 3. Digamos que para um determinado gene n no tempo m, o Kmeans retornou 2 e o CoKmeans retornou 1. Assim, o produto $2\times 1=2$, e $1^2 \leq 2 < (1+1)^2$, ou seja, $x=1$. Dessa maneira, é atribuído o valor $x=1$ para a expressão desse gene. Na Tabela 4.1 [32] temos os possíveis produtos para $k+1=3$.

Pela definição do algoritmo BiKmeans, x varia de 1 a k. Para $k=2$, os valores que caem no primeiro intervalo recebem o valor de $x=1$ e no segundo intervalo $x=2$. Entretanto, neste trabalho, o primeiro intervalo foi adaptado para corresponder ao valor $x=0$ e o segundo intervalo ao valor $x=1$. Assim discretizamos os dados de expressão gênica em 2 níveis: 0 indicando a não expressão do gene e 1 indicando a expressão do mesmo.

4.3.2 Normalização

Os valores do *combined score*, $c(x_i,x_j)$ para o par de genes x_i e x_j, obtidos através do banco de dados STRING foram normalizados para que tenhamos $0 \leq c(x_i,x_j) \leq 1$. Para isso, simplesmente dividimos cada *score* pelo maior valor encontrado entre todos os *scores*:

$$c(x_i,x_j) = \frac{c(x_i,x_j)}{\max\limits_{x,y\in X}\{c(x,y)\}}\,. \tag{4.2}$$

Algoritmo 1 BiKmeans

Entrada:

$S[n, m]$ = matriz com n genes e m amostras de expressão gênica de dados contínuos

k = número de intervalos

Saida:

$M[n, m]$ = matriz com n genes e m amostras de expressão gênica de dados discretos $0, 1$

Procedimento $BiKmeans(S[n, m], k)$

$k \leftarrow 2$
$D_l[n, m] \leftarrow \emptyset$
$D_c[n, m] \leftarrow \emptyset$
para todo $i < n$ **faça**
    $D_l[i, *] \leftarrow Kmeans(S[i, *], k + 1)$
fim para
para todo $j < m$ **faça**
    $D_c[*, j] \leftarrow CoKmeans(S[*, j], k + 1)$
fim para
para todo $i < n$ **faça**
    para todo $j < m$ **faça**
        $x \leftarrow 1$
        enquanto $x \leq k$ **faça**
            se $(D_l[i, j] * D_c[i, j]) \geq x^2$ **e** $(D_l[i, j] * D_c[i, j]) < (x + 1)^2$ **então**
                $M[i, j] \leftarrow x - 1$
            fim se
            $x \leftarrow x + 1$
        fim enquanto
    fim para
fim para
devolve $M[n, m]$

Fim Procedimento

4.4 Algoritmo de Inferência

A inferência de GRN, como proposta neste trabalho, pode ser vista como um problema de seleção de características. Dado um gene alvo y busca-se pelo melhor subconjunto de genes $Z \subseteq X$ de acordo com uma função critério. Seja $\mathcal{X}_d$ o conjunto de todos os possíveis subconjuntos de tamanho d, onde d é o número de genes preditores desejado (genes preditores são aqueles que probabilisticamente possuem grande quantidade de informação associada aos valores de um gene alvo y), e seja $J(Z)$ a função critério que avalia o subconjunto de preditores $Z \in \mathcal{X}_d$. Sem perda de generalidade, considere que quanto maior o valor de J melhor é esse subconjunto de preditores. Assim podemos formular o problema de seleção de características como: encontrar um subconjunto Z_d tal que:

$$J(Z_d) = \max_{Z \in \mathcal{X}_d} J(Z)\,, \tag{4.3}$$

onde definimos a função critério por

$$J(Z) = \omega_1 \theta(y, Z) + \omega_2 \iota(y, Z) + \omega_3 \sigma(y, Z)\,, \tag{4.4}$$

onde ω_1, ω_2 e ω_3 são pesos tal que $\omega_1 + \omega_2 + \omega_3 = 1$ e $\theta(y, Z)$ é o coeficiente de determinação para o gene y dado Z, $\iota(y, Z)$ é a informação mútua entre y e Z e $\sigma(y, Z)$ é o Conhecimento Biológico entre o gene y e o conjunto Z.

De forma geral, podemos combinar a medidas entre o CoD, IM e CB. Entretanto, como já descrito em nas Seções 2.2.4 e 2.2.4, as medidas θ e ι possuem penalizações, assim foram implementadas neste trabalho as seguintes medidas:

- Coeficiente de Determinação (θ);
- Coeficiente de Determinação com Penalização nas instâncias Não observadas (θ');
- Coeficiente de Determinação com Penalização nas instâncias Raramente observadas (θ'');
- Informação Mútua (ι);
- Informação Mútua com Penalização nas instâncias Não observadas (ι');
- Informação Mútua com Penalização nas instâncias Raramente observadas (ι'');
- Entropia de Tsallis com Penalização nas instâncias Não observadas (ι'_q);
- Conhecimento Biológico do STRING DB (σ).

Cada uma dessas medidas podem ser combinadas na função critério $J(Z)$ (eq. 4.4), respeitando a seguinte restrição: apenas uma medida de CoD (seja ela normal ou com penalização), apenas uma de IM seja ela normal ou com penalização), e opcionalmente uma de CB. Por exemplo, $J(Z) = \omega_1 \theta''(y, Z) + \omega_2 \iota(y, Z)$ é permitido, já $J(Z) = \omega_1 \theta''(y, Z) + \omega_2 \theta'$ não é permitido.

O algoritmo de seleção de características utilizado como base para este trabalho foi o IFFS (Seção 2.2.3). Aqui definimos os passos desse algoritmo, como segue descrito no Algoritmo 2. Assumindo que os dados do STRING já foram obtidos e todos os dados de entrada já foram preprocessados. Temos então $X = \{x_1, x_2, \ldots, x_n\}$ o conjunto de n genes e S o conjunto de dados de expressão gênica com m amostras. Assim desejamos obter um subconjunto $Z \in X$ de d genes preditores, de

forma que observando os valores dos preditores no tempo $t+1$ para um gene alvo y observado no tempo t, maximize a função critério $J(Z)$. Assuma também que os pesos $\omega_1 + \omega_2 + \omega_3 = 1$ e as medidas $\theta(y, Z)$, $\iota(y, Z)$ e $\sigma(y, Z)$, já foram informadas para configuração de J.

Algoritmo 2 Algoritmo de Inferência

Entrada:

$X = \{x_1, x_2, \ldots, x_n\}$ = o conjunto de n genes
$S[n, m]$ = matriz com n genes e m amostras de expressão gênica
d = número de preditores que deseja inferir para cada gene.
J = função critério.

Saida: $L[n, d+1]$ = matriz com n linhas, representando os genes de X e d preditores mais seu *score*, representando o número de colunas

Algoritmo *Inferência_de_GRN*$(X, S[n, m], d, J)$

para todo $y \in X$ **faça**
  $Z \leftarrow \emptyset$;
  enquanto VERDADE **faça**
    $SFS(y, Z, X)$; /*Adiciona mais um gene preditor ao subconjunto $Z \in X$ com o SFS*/
    se $|Z| = d$ **então**
      $L[y, *] \leftarrow Z$;
      $L[y, d+1] \leftarrow J(y, Z)$;
      PARE;
    fim se
    enquanto VERDADE **faça**
      $AUX \leftarrow SBS(y, Z)$ /*AUX recebe Z menos um gene removido pelo SBS*/
      se $J(y, AUX) > J(y, Z-1)$ **então**
        $Z \leftarrow AUX$;
        CONTINUE;
      fim se
      se $J(y, Z - z_i + x_j) > J(y, Z)$ **então** /*Se Z com o gene z_i substituído pelo gene x_j, possuir J maior do que apenas $J(Z)$ então substitua*/
        $Z \leftarrow Z - z_i$
        $Z \leftarrow Z + x_j$
      senão
        PARE;
      fim se
    fim enquanto
  fim enquanto
fim para
devolve L
Fim Algoritmo

4.5 Pós-processamento dos dados

Nesta seção será detalhado os pós-processamentos realizados no resultado obtido pelo algoritmo de seleção de características IFFS.

Sejam $L_1, L_2, \ldots, L_k$ as listas obtidas pela execução do Algoritmo 2 para $d = 1, 2, \ldots, k$. O pós-processamento apresentado no Algoritmo 3 é executado para determinar, para cada gene em X, qual o melhor subconjunto de genes preditores dentre os k possíveis.

De forma geral, podemos dizer que o Algoritmo 3 possui os seguintes passos:

Algoritmo 3 Pós-Processamento

Entrada:

$L_1[n, 2]$ = matriz com n linhas, representando os genes e $d = 1$ preditores mais seu *score*, representando o número de colunas

$L_2[n, 3]$ = matriz com n linhas, representando os genes e $d = 2$ preditores mais seu *score*, representando o número de colunas

...

$L_k[n, d + 1]$ = matriz com n linhas, representando os genes e d preditores mais seu *score*, representando o número de colunas

Saida:

$L_r[r, 3]$ = matriz com r linhas, representando número de pares de genes (arestas) e 2 (par) de genes mais seu *score*, representando o número de colunas

Procedimento $Pos\_Processamento(L_1[n, 2], L_2[n, 3], \ldots, L_k[n, d + 1])$

para todo $i < n$ **faça**
    $u \leftarrow 0$
    repete $u \leftarrow u + 1$
    até $u < k$ **e** $L_u[i, (u + 1)] < L_{u+1}[i, (u + 2)]$
    se $L_u[i, (u + 1)] < L_{u+1}[i, (u + 2)]$ **então** $L_t[i, *] \leftarrow L_{u+1}[i, *]$
    senão$L_t[i, *] \leftarrow L_u[i, *]$
    fim se
fim para
$x \leftarrow 0$
para todo $i < n$ **faça**
    $cols \leftarrow (ncols\ de\ L_t[i, *])$
    $y \leftarrow 0$
    para todo $j < cols$ **faça**
        $L_r[x, y] \leftarrow L_t[i, j]$
        $y \leftarrow y + 1$
        $L_r[x, y] \leftarrow i$
        $y \leftarrow y + 1$
        $L_r[x, y] \leftarrow L_t[i, cols]$
        $x \leftarrow x + 1$
    fim para
fim para
$Order\_Maior\_para\_Menor(L_r)$
$OPCIONAL : plot\_graph(L_r)$
devolve L_r

Fim Procedimento

Arestas	Inferida em B	Não inferida em B
Presente em A	TP	FN
Ausente em A	FP	TN

Tabela 4.2: *Matriz de confusão: TP = verdadeiro positivo; FP = falso positivo; FN = falso negativo; TN = verdadeiro negativo.*

- **Passo 1 (linha 6 - 13):** seleciona o conjunto de preditores que possui maior *score*; Em caso de empate entre o conjunto de tamanho d e $d+1$, selecione os preditores da lista de menor d.
- **Passo 2 (linha 14 - 26):** com os preditores selecionado para cada gene, mapeia as arestas, as quais neste trabalho são direcionadas. Portanto se um gene x_i ficou com 3 preditores x_j, x_k, x_l, serão mapeados as seguintes arestas $(x_j, x_i); (x_k, x_i); (x_k, x_i)$ e para cada aresta o *score* será o mesmo.
- **Passo 3 (linha 27):** a lista de arestas é ordenada em ordem não crescente de acordo com seus *score*.
- **Passo 4 (linha 28):** Opcionalmente, pode-se gerar o grafo dessa lista.

4.6 Validação

Para validar este problema de inferência de GRN é complicado, porque como verificar se a rede inferiu as arestas corretamente, sendo que não se conhece qual a rede verdadeira?. Assim, uma das formas para validar e analisar nossa metodologia foi utilizar dados sintéticos (gerados computacionalmente e não obtido de um experimento biológico) onde se conhece a rede verdadeira e tem como medir quão boa foi a recuperação das arestas inferidas por nossa metodologia. E para avaliar o desempenho da metodologia em dados extraído a partir de experimentos biológicos, consideramos como rede verdadeira aquela rede estabelecida e concordada por diversos estudos e experimentos de outros pesquisadores.

Assim sendo, Assuma que A é a rede verdadeira e B é a rede de inferida representada como um grafo direcionado. Então é possível completar as entradas da matriz de confusão [63] de acordo com a Tabela 4.2 onde:

- TP (verdadeiro positivo), é quando uma aresta é positiva e é classificado como positiva.
- FN (falso negativo), é quando uma aresta é positiva e é classificado como negativa.
- TN (verdadeiro negativo), é quando uma aresta é negativa e é classificado como negativa.
- FP (falso positivo), é quando uma aresta é negativa e é classificado como positiva.

A partir da lista ordenada pelo *score*, uma rede com e arestas é obtida considerando somente as e primeiras arestas, Assim é possível calcular várias métricas de performance para $e = 1, 2, \ldots, T$, onde $T = n(n-1)$ é o número total de arestas possíveis em uma rede de n nós. A partir disso, obtêm-se:

Taxa verdadeiro positivo:

$$\mathrm{TPR}(e) = \frac{\mathrm{TP}(e)}{\mathrm{TP}(e) + \mathrm{FN}(e)}\,, \tag{4.5}$$

Taxa falso positivo:

$$\mathrm{FPR}(e) = \frac{\mathrm{FP}(e)}{\mathrm{FP}(e) + \mathrm{TN}(e)}\,. \tag{4.6}$$

Precisão:

$$\mathrm{PRE}(e) = \frac{\mathrm{TP}(e)}{\mathrm{TP}(e) + \mathrm{FP}(e)}\,. \tag{4.7}$$

Recall:

$$\mathrm{REC}(e) = \frac{\mathrm{TP}(e)}{\mathrm{TP}(e) + \mathrm{FN}(e)} \tag{4.8}$$

Para validar as redes inferidas calculou-se a área sob a curva *Receiver Operating Characteristic (ROC)* (AUROC) e a área sob a curva *Precision-Recall* (AUPR). O motivo da escolha desses métodos de validação é porque além da curva ROC ser bastante conceituada para comparações de semelhança para GRN, essas duas áreas são utilizadas como medida de desempenho nas competições DREAM as quais são bem conceituadas e inclusive disponibilizam o *script* em Matlab para o cálculo dos mesmos. Assegurando ainda mais a validação e confiabilidade nos resultados obtidos.

A curva ROC é o gráfico da função onde o eixo x é a taxa de falso positivo (FPR ou especificidade) e o eixo y é a taxa de verdadeiro positivo (TPR ou sensibilidade) no domínio de $e = 1, 2, \ldots, T$. E a curva *Precision-Recall* representa a relação entre o *Recall* (eixo-x) e a *precision* (eixo-y). Assim, o AUROC e o AUPR é um número que resume essa relação conforme o parâmetro e é variado [39], veja exemplos dessas duas curvas nas Figuras 4.3 e 4.4.

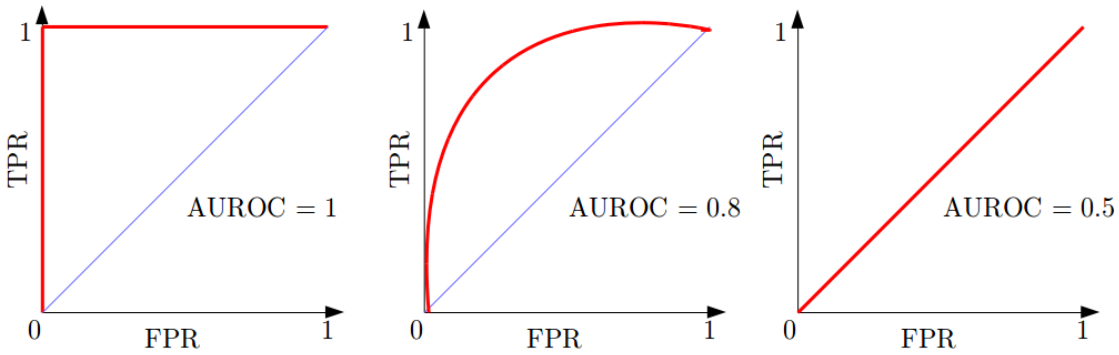


Figura 4.3: *Exemplo de curva ROC e AUROC.*

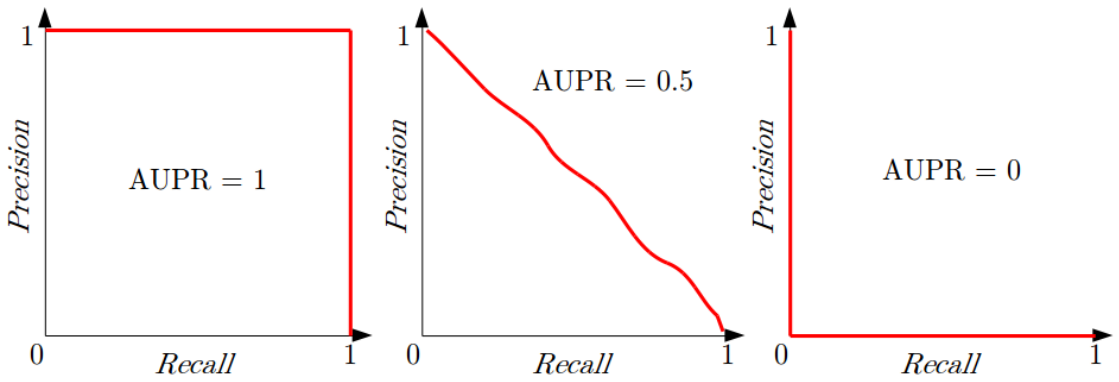


Figura 4.4: *Exemplo de curva PR e AUPR.*

De forma simplificada, uma alta *precision* implica que o algoritmo obteve substancialmente mais resultados relevantes do que irrelevantes, enquanto que um alto *recall* significa que o algoritmo retornou a maioria dos resultados relevantes e a curva PR mostra a relação entre essas duas taxas. Já a curva ROC mostra como o número de instâncias corretamente classificadas como positiva varia conforme o número de instâncias incorretamente classificadas como positivas, é a relação da sensibilidade pela especificidade [10]. Sensibilidade mede a proporção das instâncias positivas que foram corretamente identificada e a especificidade mede a proporção das instâncias positivas que foram incorretamente identificada.

Capítulo 5

Experimentos e Validação

Os experimentos realizados para teste e validação da metodologia proposta foram com os dados do DREAM challenge (*Dialogue for Reverse Engineering Assessments and Methods*) [14] e de um estudo do ciclo celular da levedura (*Saccharomyces cerevisiae*) [48].

Nas competições DREAM a GRN é conhecida, então a precisão da inferência pode ser avaliada e para a levedura, a rede tida como verdadeira foi retirada do banco de dados KEGG.

Nas seções seguintes relatamos os resultados mais significativos obtidos por nossa metodologia. No Apêndice A colocamos todos os testes e resultados obtidos. Como descrito na Seção 4.2, executamos o algoritmo de seleção de características IFFS para $k = 4$, ou seja, para $d = 1, \ldots, 4$. Assim, para cada experimento realizado selecionou 4 listas com, respectivamente, $d = 1, \ldots, 4$ preditores para cada gene de X. Após o término da execução, foi realizado o pós-processamento (ver Algoritmo 3) para obtenção da lista de arestas inferidas com seus respectivos *scores* ordenados de forma não crescente. Com as arestas inferidas e ordenadas do maior para o menor *score*, calculamos o AUROC e o AUPR através de um *script* em MATLAB fornecido pela organização da competição DREAM5, já que as mesmas formas de validação foram utilizadas por ela. Assim, bastou adaptar o *script* do DREAM5 para também validar os dados do DREAM4 e da levedura. Os dados e resultados podem ser conferidos logo a seguir.

5.1 DREAM4

Nesta competição, consideramos apenas as redes de séries temporais *in silico* de tamanho 10 e 100. Há 5 redes de tamanho 10, cada uma com 5 diferentes experimentos de 21 instantes de tempo. E há 5 redes de tamanho 100, cada uma com 10 diferentes experimentos de 21 instantes de tempo. Os resultados podem ser conferido nas Tabelas 5.1 e 5.2. Aplicamos a função critério $J(Z)$ baseada apenas em CoD e IM já que os dados são sintéticos. Para a medida de penalização utilizamos $\beta = 0.8$, $\alpha = 1$ e para a entropia de Tsallis utilizamos $q = 2.5$ de acordo com os valores testados respectivamente em [33] e [34].

Para todas as tabelas deste capítulo, as células destacadas em azul indicam o melhor valor de AUROC e as destacadas em verde indicam a pontuação mais alta para os valores de AUPR de uma determinada rede.

Observando as Tabelas 5.1 e 5.2 com o desempenho individual de cada medida temos que para 3 redes da Tabela 5.1, a medida de CoD obteve maior AUROC o que implica em um melhor desempenho, entretanto para as 2 redes restantes da Tabela 5.1 e para todas as redes da Tabela 5.2,

	Rede 1		Rede 2		Rede 3		Rede 4		Rede 5	
Medida	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**
$\theta(y, Z)$	0.65	0.39	0.65	0.34	0.63	0.29	0.64	0.24	0.75	0.37
$\theta'(y, Z)$	0.63	0.39	0.56	0.28	0.68	0.40	0.60	0.25	0.65	0.27
$\theta''(y, Z)$	0.55	0.25	0.41	0.17	0.54	0.21	0.65	0.24	0.60	0.24
$\iota(y, Z)$	0.58	0.40	0.51	0.22	0.61	0.25	0.64	0.33	0.66	0.28
$\iota'(y, Z)$	0.63	0.44	0.61	0.27	0.48	0.21	0.50	0.19	0.51	0.15
$\iota''(y, Z)$	0.62	0.35	0.54	0.22	0.67	0.31	0.67	0.34	0.53	0.18
$\iota'_q(y, Z)$	0.59	0.34	0.53	0.23	0.69	0.40	0.68	0.36	0.72	0.29

Tabela 5.1: *AUROC e AUPR para as redes do DREAM4 de tamanho 10 medindo o desempenho individual de cada medida.*

	Rede 1		Rede 2		Rede 3		Rede 4		Rede 5	
Medida	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**
$\theta(y, Z)$	0.59	0.11	0.53	0.09	0.55	0.08	0.53	0.07	0.55	0.09
$\theta'(y, Z)$	0.56	0.09	0.52	0.08	0.54	0.07	0.52	0.07	0.52	0.08
$\theta''(y, Z)$	0.57	0.10	0.50	0.07	0.55	0.07	0.53	0.07	0.54	0.09
$\iota(y, Z)$	0.60	0.16	0.53	0.09	0.57	0.12	0.56	0.11	0.57	0.12
$\iota'(y, Z)$	0.55	0.07	0.53	0.09	0.54	0.08	0.54	0.08	0.52	0.06
$\iota''(y, Z)$	0.60	0.16	0.54	0.10	0.57	0.12	0.55	0.10	0.55	0.11
$\iota'_q(y, Z)$	0.60	0.15	0.54	0.10	0.56	0.10	0.56	0.10	0.56	0.11

Tabela 5.2: *AUROC e AUPR para as redes do DREAM4 de tamanho 100 medindo o desempenho individual de cada medida.*

a IM obteve os melhores valores para AUROC e AUPR. Note também que no valor do AUPR da Rede 1 da Tabela 5.1, a IM obteve um valor superior ao do AUPR do CoD, mesmo o CoD possuindo melhor AUROC. Isso pode acontecer, pois o AUROC e o AUPR como já descrito na Seção 4.6, são funções que medem diferentes relações. Assim um algoritmo que melhora o AUROC não garante que também vai melhorar o AUPR e isso está provado em [10].

O diferencial deste trabalho foi que não adaptamos o algoritmo de inferência aos conjuntos de dados com informações adicionais e específicas que o DREAM4 fornece. A justificativa para isso é que essas informações raramente são disponíveis em experimentos biológicos.

O propósito do trabalho é fornecer um conjunto de funções e comparar o desempenho de cada medida utilizando apenas os dados de séries temporais.

Veja na Figura 5.1 a GRN tida como verdade da Rede 4 com 10 genes e na Figura 5.2 a GRN inferida utilizando $J(Z) = \theta''(y, Z) + \iota''(y, Z)$, a qual obteve 0.8 de AUROC e 0.47 de AUPR, nossa melhor inferência para esse conjunto de dados.

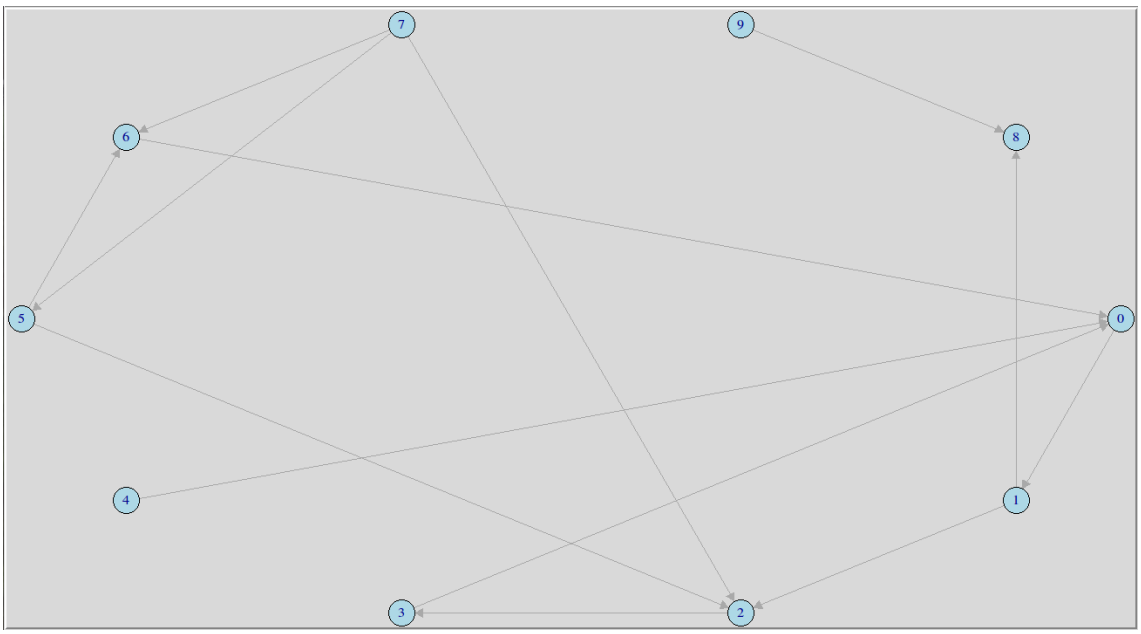


Figura 5.1: *Grafo da GRN tida como verdade da Rede 4 com 10 genes do DREAM4.*

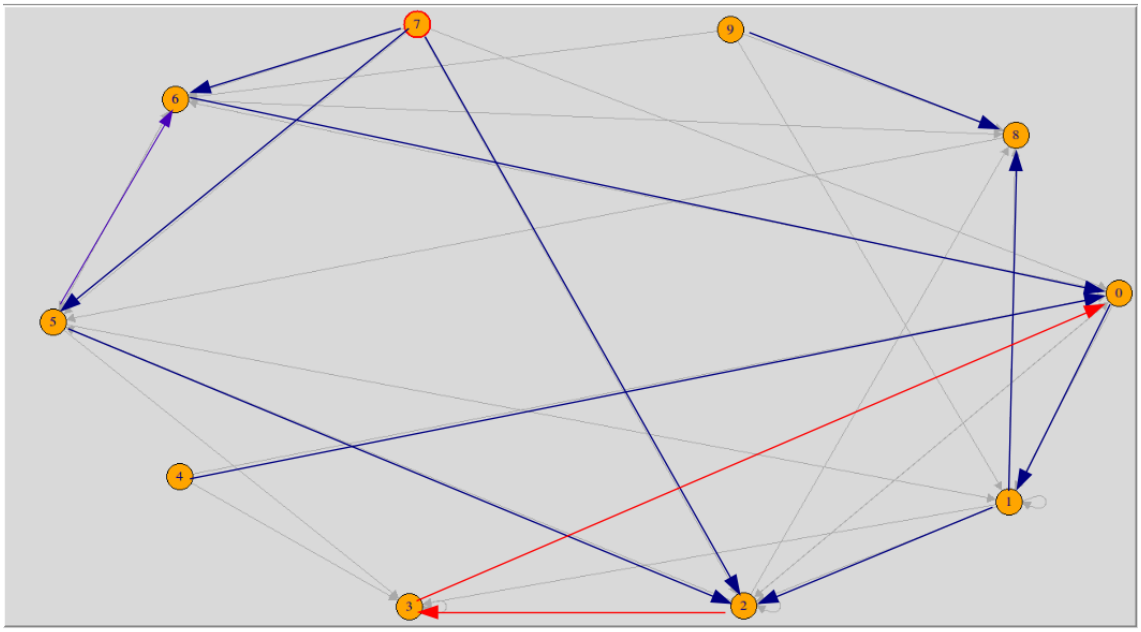


Figura 5.2: *Grafo da GRN inferida da Rede 4 com 10 genes do DREAM4. As arestas destacadas em azul são todas as arestas corretamente inferidas, as destacadas em vermelho são as arestas não inferidas mas que existem no grafo verdadeiro e as arestas não destacadas são as arestas inferidas incorretamente.*

Comparando os grafo da rede verdadeira (Figura 5.1) com o grafo da rede inferida (Figura 5.2), temos que das 13 arestas existentes, 11 arestas foram recuperadas e apenas duas não foram inferidas.

E apesar de 13 arestas terem sido inferidas a mais (incorretamente) grande parte das arestas relevantes (as verdadeiras) foram inferidas corretamente, obtendo assim um bom valor para o AUROC e AUPR. Veja na Figura 5.3 o gráfico das curvas ROC e PR deste teste.

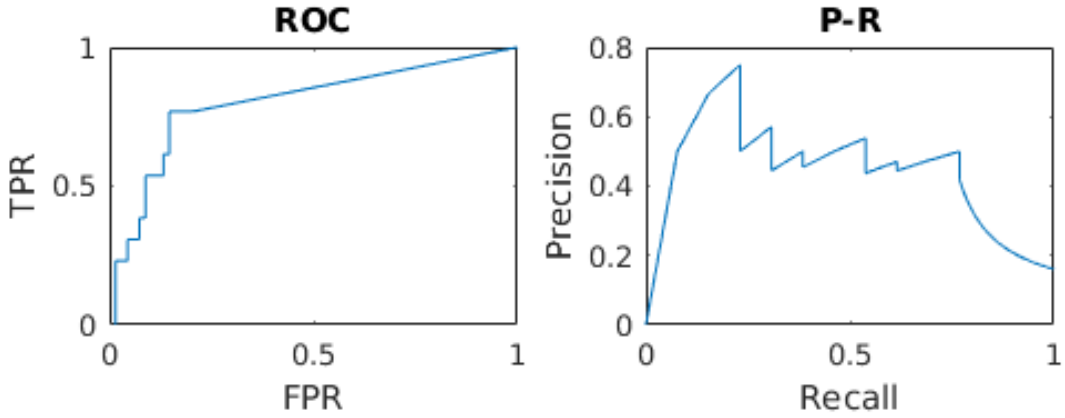


Figura 5.3: *À esquerda temos o gráfico da curva ROC e à direita o gráfico da curva PR. Comparando essas curvas obtidas com as possíveis curvas do gráfico ROC (ver Figura 4.3) e PR (ver Figura 4.4), podemos observar que, as curvas apresentaram um comportamento mais perto da curva máxima do que da curva mínima, sendo um bom resultado.*

5.2 DREAM5

Esse desafio consiste em 4 diferentes redes: Rede 1 = *in silico*, Rede 2 = *Staphylococcus aureus*, Rede 3 = *Escherichia coli* and Rede 4 = *Saccharomyces cerevisiae*. A Rede 2 não foi testada porque ainda não se tem estabelecido a rede verdadeira para ela.

A primeira motivação para utilizar os dados desse desafio foi porque pensamos em consultar o STRING DB e adicionar informação biológica para inferir as redes, já que as Redes 2, 3 e 4 são de organismos biológicos. No entanto não foram disponibilizados os nomes dos genes no formato de *locus name*, apenas em índices (número) o que não permitiu a busca em dados do STRING já que para consultar o STRING necessita do *locus name*. Mesmo assim, testamos nossa metodologia para comparar as performance das medidas e os resultados estão na Tabela 5.3. Para este conjunto de teste, temos a Rede 1 com 1643 genes e 487 instantes de tempo, Rede 2 com 4511 genes e 487 instantes de tempo e a Rede 3 com 5950 genes e 321 instantes de tempo.

	Rede 1		**Rede 3**		**Rede 4**	
Medida	**AUROC**	**AUPR**	**AUROC**	**AUPR**	**AUROC**	**AUPR**
$\theta(y, Z)$	0.52	0.03	0.50	0.01	0.50	0.02
$\theta'(y, Z)$	0.52	0.03	0.50	0.01	0.50	0.02
$\theta''(y, Z)$	0.51	0.02	0.50	0.01	0.50	0.02
$\iota(y, Z)$	0.54	0.04	0.50	0.01	0.50	0.02
$\iota'(y, Z)$	0.52	0.03	0.50	0.01	0.50	0.02
$\iota''(y, Z)$	0.54	0.04	0.50	0.01	0.50	0.02
$\iota'_q(y, Z)$	0.54	0.04	0.50	0.01	0.50	0.02

Tabela 5.3: *AUROC e AUPR para as redes do DREAM5 medindo o desempenho individual de cada medida.*

Novamente, neste experimento (Tabela 5.3) pode-se perceber que a IM obteve uma performance um pouco melhor do que o CoD. Os valores de AUROC e AUPR não são muito bons, entretanto

vale ressaltar que neste teste o número de genes é muito maior do que o número de amostras e não foi possível utilizar nenhum conhecimento biológico e também não utilizamos as informações extras e específicas que o DREAM5 disponibilizou.

5.3 Ciclo-Celular da Levedura

Finalmente, neste experimento foi possível adicionar informação biológica a função $J(Z)$. Os parâmetros α e β continuaram os mesmos dos experimentos do DREAM. O conjunto de dados de expressão gênica S do ciclo celular da levedura *Saccharomyces cerevisiae* foram retirados do trabalho do Spellman [48], o qual realizou 6 experimentos, chamados de cln3, clb2, alpha, cdc15, cdc28 e elutriation. Os dois primeiro experimentos nós descartamos porque cada um continha apenas 2 instantes de tempo, assim permanecemos com os 4 últimos, onde concatenamos os seguintes instantes de tempo: alpha com 18 instante de tempo, cdc15 com 24 instantes de tempo, cdc28 com 17 instantes de tempo e elutriation com 14 instantes de tempo.

Para validar essa metodologia, a rede tida como verdadeira foi obtida a partir dos dados do KEGG [25] para a rede do ciclo celular do mesmo organismo, a qual possui 126 genes. Assim, dos 6179 genes disponibilizados pelo trabalho do Spellman [48], realizamos uma intersecção com os 126 genes do KEGG, assim selecionamos os dados de expressão gênica da levedura para esses genes. Portanto, para este conjunto de teste nós temos somente uma rede com 126 genes e 73 instantes de tempo. E o resultado pode ser verificado nas Tabelas 5.4 e 5.5.

	Rede 1	
Medida	**AUROC**	**AUPR**
$\theta(y, Z)$	0.5	0.03
$\theta'(y, Z)$	0.5	0.03
$\theta''(y, Z)$	0.5	0.03
$\iota(y, Z)$	0.5	0.03
$\iota'(y, Z)$	0.51	0.03
$\iota''(y, Z)$	0.5	0.03
$\iota'_q(y, Z)$	0.51	0.04
$\sigma(y, Z)$	0.54	0.04

Tabela 5.4: *AUROC e AUPR para a rede da levedura medindo o desempenho individual de cada medida.*

Nesta Tabela 5.4, podemos notar novamente que a IM obteve um desempenho levemente superior ao CoD e o CB (Conhecimento Biológico) obteve um resultado ainda melhor.

	Rede 1	
Medida	**AUROC**	**AUPR**
$\theta(y,Z)+\iota(y,Z)$	0.51	0.04
$\theta(y,Z)+\iota'_q(y,Z)$	0.5	0.03
$\theta(y,Z)+\sigma(y,Z)$	0.55	0.06
$\theta(y,Z)+\iota(y,Z)+\sigma(y,Z)$	0.55	0.05
$\theta(y,Z)+\iota'_q(y,Z)+\sigma(y,Z)$	0.56	0.06
$\theta'(y,Z)+\iota'(y,Z)$	0.5	0.03
$\theta'(y,Z)+\iota'_q(y,Z)$	0.05	0.03
$\theta'(y,Z)+\sigma(y,Z)$	0.56	0.06
$\theta'(y,Z)+\iota'(y,Z)+\sigma(y,Z)$	0.56	0.06
$\theta'(y,Z)+\iota'_q(y,Z)+\sigma(y,Z)$	0.56	0.06
$\theta''(y,Z)+\iota''(y,Z)$	0.5	0.03
$\theta''(y,Z)+\iota''_q(y,Z)$	0.5	0.03
$\theta''(y,Z)+\sigma(y,Z)$	0.56	0.05
$\theta''(y,Z)+\iota''(y,Z)+\sigma(y,Z)$	0.56	0.06
$\theta''(y,Z)+\iota''_q(y,Z)+\sigma(y,Z)$	0.56	0.06
$\iota(y,Z)+\sigma(y,Z)$	0.56	0.06
$\iota'(y,Z)+\sigma(y,Z)$	0.58	0,07
$\iota''(y,Z)+\sigma(y,Z)$	0.56	0.06
$\iota'_q(y,Z)+\sigma(y,Z)$	0.56	0.07

Tabela 5.5: *AUROC e AUPR para a rede da levedura medindo o desempenho combinado das medidas. Repare que $\sigma(y,Z)$ está destacado em azul para indicar que onde ele foi incorporado ajudou a obter um desempenho melhor do que cada medida individualmente obteve e melhor do que as combinações onde ele não foi incorporado.*

Na Tabela 5.5, combinamos duas medidas na função critério $J(Z)$ com a seguinte distribuição de peso: $\omega_i = 0.5$ e $\omega_j = 0.5$ para $i = \{1,2,3\}$ e $j = \{1,2,3\}$ e para combinação de 3 medidas utilizamos: $\omega_1 = 0.3$, $\omega_2 = 0.3$ e $\omega_3 = 0.4$. Observe todas as combinações que incorporam o CB ($\sigma(y,Z)$), que apesar de sutilmente, o CB contribuiu de forma positiva na inferência da GRN. Apesar dos valores de AUROC e AUPR não serem tão altos, conseguimos comparar essas medidas e constatar que o CB *a priori* é um bom recurso a ser usado.

5.4 Dificuldades Enfrentadas

Durante o desenvolvimento do trabalho enfrentamos diversos fatos que prejudicaram nossos testes, por exemplo:

- O problema da Inferência de GRN a partir de dados temporais de expressão gênica ser um problema inverso mal posto.
- O fato de estarmos inferindo apenas uma GRN e não diversas possibilidades de GRN, ou seja, mais de uma GRN como candidatas a GRN verdadeira.
- Não possuir o *locus name* dos genes da competição DREAM5, para poder incorporar CB e

inferir uma GRN melhor visto o número de genes ser mais que 3 vezes superior ao número de amostras.

- A dificuldade em conseguir o conjunto de dados de expressão gênica obtido em série temporal de um determinado organismo e relacionado a algum processo celular.

Capítulo 6

Discussão e Conclusão

Neste trabalho, nós propusemos uma metodologia para a inferência de redes de regulação gênica baseada na abordagem de seleção de características, usando o algoritmo IFFS. Além disso, projetamos a função critério para incluir duas medidas comumente usadas para inferência de GRN (coeficiente de determinação, informação mútua e suas variações). Também analisamos o desempenho da metodologia quando incorpora o conhecimento biológico *a priori* fornecido pelo banco de dados STRING.

Os dados das competições DREAM serviram para validar esta metodologia. Comparando nossos resultados com os dos demais times, conseguiríamos a posição 24º de 29 para as redes de 10 genes (DREAM4), a 16º posição para as redes de 100 genes (DREAM4) e no DREAM5 conseguiríamos o 33º lugar de 35. Não parece ser uma colocação muito boa, entretanto vale ressaltar que utilizamos apenas os dados de séries temporais enquanto os demais time provavelmente devem ter utilizado as informações adicionais e específicas de cada conjunto de dados. Que no DREAM4 foram: genes que sofreram *knockouts*, *knockdowns* e os que são multifatoriais. E para o DREAM5 foram: genes que sofreram pertubação e os respectivos níveis de pertubações, genes tratados, genes deletados e genes sobre expresso.

Como sabemos que em experimentos reais nem sempre se obtém nada além dos dados temporais das expressões dos genes, preferimos manter-nos nesse padrão comum, porque além dos dados adicionais nem sempre serem disponíveis, provavelmente também não seguem um padrão. Por exemplo os dados adicionais do DREAM4 para o DREAM5 são totalmente diferentes um do outro, assim seria necessário um programa específico para cada caso. E como no nosso caso não queremos nos especificar e sim generalizar para poder disponibilizar essa ferramenta em R que poderá ser utilizada não somente para reconstrução de GRN, mas também para outros propósitos como reconhecimento de padrão, classificação, entre outros, assim, preferimos utilizar apenas os dados temporais das expressões dos genes.

Como resultado geral da nossa metodologia, podemos considerar que obtivemos resultados razoáveis e compreensíveis, visto as circunstâncias em que operamos e as dificuldade que enfrentamos. Como conclusão do trabalho desenvolvido, observamos que a IM desempenhou melhor do que o coeficiente de determinação e quando se tinha CB disponível, o mesmo contribuiu de forma positiva a inferir a GRN.

Acredito que os objetivos deste trabalho foram alcançados e os resultados obtidos condizem com alguns relatos na literatura, assim deixamos a nossa contribuição do pacote Measures_GRN em R, brevemente disponível com toda documentação completa em https://github.com/camila-koike/

Measure_GRN.

6.1 Sugestões para Pesquisas Futuras

Como sugestões para trabalhos futuros, pretendemos explorar como usar várias fontes de conhecimento biológico *a priori* em vez de apenas o banco de dados STRING. Neste trabalho, nós inferimos apenas a topologia da rede, mas também é possível inferir as funções Booleanas da rede, e analisar sua dinâmica para pequenas redes. Podemos também usar esta metodologia para conjuntos de dados de organismo que possui algum conhecimento biológico disponível, mas que não foram tão estudados como no caso da levedura, e gerar hipóteses de GRN para ser testadas em *wetlab*.

Apêndice A

Testes e Resultados Individuais

Este apêndice contém todos os testes realizados para os conjuntos de dados do DREAM4, DREAM5 e da levedura.

	Network 1		
Medida	**AUROC**	**AUPR**	**TEMPO**
$\theta(y, Z)$	0,5	0,03	0.80 segs
$\theta'(y, Z)$	0,5	0,03	0.84 segs
$\theta''(y, Z)$	0,5	0,03	0.84 segs
$\iota(y, Z)$	0,5	0,03	1.02 segs
$\iota'(y, Z)$	0,51	0,03	0.96 segs
$\iota''(y, Z)$	0,5	0,03	0.90 segs
$\iota'_q(y, Z)$	0,51	0,04	1.40 segs
$\sigma(y, Z)$	0,54	0,04	0.11 segs
$\theta(y, Z) + \iota(y, Z)$	0,51	0,04	1.79 segs
$\theta(y, Z) + \iota'_q(y, Z)$	0,5	0,03	2.13 segs
$\theta(y, Z) + \sigma(y, Z)$	0,55	0,06	0.88 segs
$\theta(y, Z) + \iota(y, Z) + \sigma(y, Z)$	0,55	0,05	1.84 segs
$\theta(y, Z) + \iota'_q(y, Z) + \sigma(y, Z)$	0,56	0,06	2.18 segs
$\theta'(y, Z) + \iota'(y, Z)$	0,5	0,03	1.63 segs
$\theta'(y, Z) + \iota'_q(y, Z)$	0,05	0,03	2.08 segs
$\theta'(y, Z) + \sigma(y, Z)$	0,56	0,06	0.91 segs
$\theta'(y, Z) + \iota'(y, Z) + \sigma(y, Z)$	0,56	0,06	1.72 segs
$\theta'(y, Z) + \iota'_q(y, Z) + \sigma(y, Z)$	0,56	0,06	2.19 segs
$\theta''(y, Z) + \iota''(y, Z)$	0,5	0,03	1.64 segs
$\theta''(y, Z) + \iota'_q(y, Z)$	0,51	0,03	2.11 segs
$\theta''(y, Z) + \sigma(y, Z)$	0,56	0,05	0.82 segs
$\theta''(y, Z) + \iota''(y, Z) + \sigma(y, Z)$	0,56	0,06	1.75 segs
$\theta''(y, Z) + \iota'_q(y, Z) + \sigma(y, Z)$	0,56	0,06	2.14 segs
$\iota(y, Z) + \sigma(y, Z)$	0,56	0,06	1.08 segs
$\iota'(y, Z) + \sigma(y, Z)$	0,58	0,07	0.94 segs
$\iota''(y, Z) + \sigma(y, Z)$	0,56	0,06	0.99 segs
$\iota'_q(y, Z) + \sigma(y, Z)$	0,56	0,07	1.44 segs

Tabela A.1: *AUROC e AUPR da levedura*

	Network 1			Network 2			Network 3			Network 4			Network 5		
Medida	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**
$\theta(y,Z)$	0.65	0.39	0.06 segs	0.65	0.34	0.03 segs	0.63	0.29	0.06 segs	0.64	0.24	0.04 segs	0.75	0.37	0.03 segs
$\theta'(y,Z)$	0.63	0.39	0.01 segs	0.56	0.28	0.01 segs	0.68	0.41	0.02 segs	0.60	0.25	0.01 segs	0.65	0.27	0.01 segs
$\theta''(y,Z)$	0.55	0.25	0.01 segs	0.41	0.17	0.01 segs	0.54	0.21	0.01 segs	0.65	0.24	0.01 segs	0.60	0.24	0.01 segs
$\iota(y,Z)$	0.58	0.40	0.01 segs	0.51	0.22	0.01 segs	0.61	0.25	0.01 segs	0.64	0.33	0.01 segs	0.66	0.28	0.01 segs
$\iota'(y,Z)$	0.63	0.44	0.01 segs	0.61	0.27	0.01 segs	0.48	0.21	0.01 segs	0.50	0.19	0.01 segs	0.51	0.15	0.01 segs
$\iota''(y,Z)$	0.62	0.35	0.01 segs	0.54	0.22	0.01 segs	0.67	0.31	0.01 segs	0.67	0.34	0.01 segs	0.53	0.18	0.01 segs
$\iota'_q(y,Z)$	0.59	0.34	0.01 segs	0.53	0.23	0.01 segs	0.69	0.40	0.02 segs	0.68	0.36	0.01 segs	0.72	0.29	0.01 segs
$\theta(y,Z)+\iota(y,Z)$	0.59	0.26	0.01 segs	0.51	0.22	0.01 segs	0.60	0.24	0.01 segs	0.61	0.27	0.01 segs	0.68	0.28	0.01 segs
$\theta'(y,Z)+\iota'(y,Z)$	0.56	0.25	0.01 segs	0.65	0.40	0.01 segs	0.68	0.34	0.01 segs	0.61	0.24	0.02 segs	0.60	0.24	0.01 segs
$\theta''(y,Z)+\iota''(y,Z)$	0.63	0.36	0.01 segs	0.51	0.22	0.01 segs	0.54	0.22	0.01 segs	0.80	0.47	0.01 segs	0.57	0.21	0.02 segs
$\theta(y,Z)+\iota'_q(y,Z)$	0.61	0.30	0.01 segs	0.53	0.23	0.02 segs	0.68	0.34	0.02 segs	0.68	0.37	0.02 segs	0.72	0.29	0.02 segs
$\theta'(y,Z)+\iota'_q(y,Z)$	0.60	0.33	0.01 segs	0.57	0.28	0.02 segs	0.65	0.31	0.01 segs	0.70	0.39	0.02 segs	0.70	0.30	0.01 segs
$\theta''(y,Z)+\iota'_q(y,Z)$	0.60	0.31	0.02 segs	0.48	0.20	0.02 segs	0.62	0.28	0.03 segs	0.72	0.35	0.02 segs	0.66	0.27	0.02 segs

Tabela A.2: *AUROC e AUPR para as redes do DREAM4 de tamanho 10 e o tempo de execução.*

	Network 1			Network 2			Network 3			Network 4			Network 5		
Medida	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**
$\theta(y,Z)$	0.59	0.11	1.17 segs	0.53	0.09	1.11 segs	0.55	0.08	1.09 segs	0.53	0.07	1.02 segs	0.55	0.09	1.00 segs
$\theta'(y,Z)$	0.56	0.09	1.08 segs	0.52	0.08	1.07 segs	0.54	0.07	1.05 segs	0.52	0.07	1.17 segs	0.52	0.08	1.05 segs
$\theta''(y,Z)$	0.57	0.10	0.99 segs	0.50	0.07	0.98 segs	0.55	0.07	0.95 segs	0.53	0.07	0.97 segs	0.54	0.09	0.96 segs
$\iota(y,Z)$	0.60	0.16	1.14 segs	0.53	0.09	1.15 segs	0.57	0.12	1.15 segs	0.56	0.11	1.24 segs	0.57	0.12	1.20 segs
$\iota'(y,Z)$	0.55	0.07	1.14 segs	0.53	0.09	1.24 segs	0.54	0.08	1.74 segs	0.54	0.08	1.15 segs	0.52	0.06	1.19 segs
$\iota''(y,Z)$	0.60	0.16	1.16 segs	0.54	0.10	1.23 segs	0.57	0.12	1.21 segs	0.55	0.10	1.16 segs	0.55	0.11	1.19 segs
$\iota'_q(y,Z)$	0.60	0.15	1.55 segs	0.54	0.10	1.65 segs	0.56	0.10	1.60 segs	0.56	0.10	1.48 segs	0.56	0.11	1.67 segs
$\theta(y,Z)+\iota(y,Z)$	0.62	0.19	2.50 segs	0.53	0.09	2.41 segs	0.56	0.10	2.33 segs	0.55	0.11	2.33 segs	0.56	0.11	0.01 segs
$\theta'(y,Z)+\iota'(y,Z)$	0.56	0.12	2.18 segs	0.53	0.11	2.31 segs	0.55	0.09	2.26 segs	0.53	0.07	2.28 segs	0.55	0.11	0.02 segs
$\theta''(y,Z)+\iota''(y,Z)$	0.58	0.15	2.44 segs	0.53	0.09	2.23 segs	0.58	0.11	2.33 segs	0.54	0.08	2.33 segs	0.55	0.10	0.01 segs
$\theta(y,Z)+\iota'_q(y,Z)$	0.60	0.15	2.64 segs	0.53	0.09	2.73 segs	0.56	0.09	2.65 segs	0.54	0.09	2.53 segs	0.55	0.10	0.02 segs
$\theta'(y,Z)+\iota'_q(y,Z)$	0.57	0.13	2.62 segs	0.52	0.08	2.63 segs	0.56	0.10	2.50 segs	0.53	0.08	2.47 segs	0.53	0.08	0.02 segs
$\theta''(y,Z)+\iota'_q(y,Z)$	0.58	0.12	2.74 segs	0.53	0.09	2.64 segs	0.58	0.12	2.71 segs	0.55	0.09	2.88 segs	0.55	0.10	0.01 segs

Tabela A.3: *AUROC e AUPR para as redes do DREAM4 de tamanho 100 e o tempo de execução.*

	Network 1			Network 3			Network 4		
Medida	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**
$\theta(y,Z)$	0.52	0.03	15.82 mins	0.50	0.01	1.94 horas	0.50	0.02	1.86 horas
$\theta'(y,Z)$	0.52	0.03	16.79 mins	0.50	0.01	1.98 horas	0.50	0.02	1.64 horas
$\theta''(y,Z)$	0.51	0.02	15.49 mins	0.50	0.01	1.97 horas	0.50	0.02	1.54 horas
$\iota(y,Z)$	0.54	0.04	16.21 mins	0.50	0.01	1.85 horas	0.50	0.02	1.87 horas
$\iota'(y,Z)$	0.52	0.03	16.30mins	0.50	0.01	1.82 horas	0.50	0.02	1.85 horas
$\iota''(y,Z)$	0.54	0.04	16.14 mins	0.50	0.01	1.69 horas	0.50	0.02	1.64 horas
$\iota'_q(y,Z)$	0.54	0.04	18.75 mins	0.50	0.01	2.05hours	0.50	0.02	2.89 horas
$\theta(y,Z)+\iota(y,Z)$	0.54	0.04	32.77 mins	0.50	0.01	3.81 horas	0.50	0.02	4.86 horas
$\theta'(y,Z)+\iota'(y,Z)$	0.52	0.03	33.46 mins	0.50	0.01	3.64 horas	0.50	0.02	4.59 horas
$\theta''(y,Z)+\iota''(y,Z)$	0.52	0.03	32.66 mins	0.50	0.01	3.76 horas	0.50	0.02	4.65 horas
$\theta(y,Z)+\iota'_q(y,Z)$	0.53	0.03	36.03 mins	0.50	0.01	3.62 horas	0.50	0.02	4.76 horas
$\theta'(y,Z)+\iota'_q(y,Z)$	0.52	0.03	36.53 mins	0.50	0.01	2.76 horas	0.50	0.02	4.25 horas
$\theta''(y,Z)+\iota'_q(y,Z)$	0.52	0.02	36.92 mins	0.50	0.01	3.04 horas	0.50	0.02	5.27 horas

Tabela A.4: *AUROC e AUPR para as redes do AVG-DREAM5 e o tempo de execução.*

	Network 1			Network 3			Network 4		
Medida	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**	**AUROC**	**AUPR**	**TEMPO**
$\theta(y,Z)$	0.53	0.03	16.54 mins	0.50	0.01	2.726822	0.50	0.02	2.92 horas
$\theta'(y,Z)$	0.52	0.03	17.81 mins	0.50	0.01	2.18 horas	0.50	0.02	2.48 horas
$\theta''(y,Z)$	0.51	0.02	16.08 mins	0.50	0.01	1.97 horas	0.50	0.02	2.22 horas
$\iota(y,Z)$	0.54	0.04	16.52 mins	0.50	0.01	2.15 horas	0.50	0.02	2.37 horas
$\iota'(y,Z)$	0.52	0.03	19.31 mins	0.50	0.01	2.48 horas	0.50	0.02	2.50 horas
$\iota''(y,Z)$	0.54	0.04	16.66 mins	0.50	0.01	2.25 horas	0.50	0.02	2.39 horas
$\iota'_q(y,Z)$	0.54	0.04	18.70 mins	0.50	0.01	2.37 horas	0.50	0.02	2.76 horas
$\theta(y,Z)+\iota(y,Z)$	0.54	0.04	34.20 mins	0.50	0.01	4.58 horas	0.50	0.02	4.65 horas
$\theta'(y,Z)+\iota'(y,Z)$	0.53	0.03	34.50 mins	0.50	0.01	4.72 horas	0.50	0.02	5.67 horas
$\theta''(y,Z)+\iota''(y,Z)$	0.53	0.03	34.29 mins	0.50	0.01	4.70 horas	0.50	0.02	6.48 horas
$\theta(y,Z)+\iota'_q(y,Z)$	0.53	0.03	38.42 mins	0.50	0.01	4.95 horas	0.50	0.02	5.04 horas
$\theta'(y,Z)+\iota'_q(y,Z)$	0.52	0.03	37.57 mins	0.50	0.01	5.59 horas	0.50	0.02	6.08 horas
$\theta''(y,Z)+\iota'_q(y,Z)$	0.52	0.02	38.78 mins	0.50	0.01	5.11 horas	0.50	0.02	6.20 horas

Tabela A.5: *AUROC e AUPR para as redes do DREAM5 e o tempo de execução.*

Referências Bibliográficas

[1] **Alberts** *et al.***(2009)** Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts e Peter Walter. *Biologia molecular da célula*. Artmed Editora. Citado na pág. 7

[2] **Alwine** *et al.***(1977)** James C Alwine, David J Kemp e George R Stark. Method for detection of specific rnas in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with dna probes. *Proceedings of the National Academy of Sciences*, 74(12):5350–5354. Citado na pág. 1, 8

[3] **Ast(2004)** Gil Ast. How did alternative splicing evolve? *Nature Reviews Genetics*, 5(10): 773–782. Citado na pág. 7

[4] **Barrera** *et al.***(2007)** J. Barrera, R. M. Cesar-Jr, D. C. Martins-Jr, R. Z. N. VÃªncio, E. F. Merino, M. M. Yamamoto, F. G. Leonardi, C. A. B. Pereira e H. A. Portillo. Constructing probabilistic genetic networks of *Plasmodium falciparum*, from dynamical expression signals of the intraerythrocytic development cycle. Em Patrick McConnell, Simon M. Lin e Patrick Hurban, editors, *Methods of Microarray Data Analysis V*, páginas 11–26. Springer US. ISBN 978-0-387-34569-7. Citado na pág. 16

[5] **Benson** *et al.***(2000)** Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, Barbara A Rapp e David L Wheeler. Genbank. *Nucleic acids research*, 28(1):15–18. Citado na pág. 2

[6] **Boltzmann(1970)** Ludwig Boltzmann. Über die beziehung eines allgemeinen mechanischen satzes zum zweiten hauptsatze der wärmetheorie. Em *Kinetische Theorie II*, páginas 240–247. Springer. Citado na pág. 17, 21

[7] **Borland** *et al.***(1998)** Lisa Borland, Angel R. Plastino e Constantino Tsallis. Information gain within nonextensive thermostatistics. *Journal of Mathematical Physics*, 39(12):6490–6501. Citado na pág. 21

[8] **Clausius(1867)** Rudolf Clausius. *The mechanical theory of heat: with its applications to the steam-engine and to the physical properties of bodies*. J. van Voorst. Citado na pág. 17

[9] **Cover e Thomas(1991)** Thomas M Cover e Joy A Thomas. Elements of information. *TheoryWiley, New York*. Citado na pág. 17, 18, 19, 20

[10] **Davis e Goadrich(2006)** Jesse Davis e Mark Goadrich. The relationship between precision-recall and roc curves. Em *Proceedings of the 23rd international conference on Machine learning*, páginas 233–240. ACM. Citado na pág. 35, 38

[11] **De Jong(2002)** Hidde De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103. Citado na pág. 2, 9

[12] **Dougherty** *et al.***(2000)** Edward R Dougherty, Seungchan Kim e Yidong Chen. Coefficient of determination in nonlinear signal processing. *Signal Processing*, 80(10):2219–2235. Citado na pág. 15

[13] **Dougherty** *et al.***(2009)** Edward R Dougherty, Marcel Brun, Jeffrey M Trent e Michael L Bittner. Conditioning-based modeling of contextual genomic regulation. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 6(2):310–320. Citado na pág. 15

[14] **DREAM(2009)** DREAM. DREAM: Dialogue for Reverse Engineering Assessments and Methods, 2009. project website: http://dreamchallenges.org/. Citado na pág. 37

[15] **Friedman** *et al.***(2000)** Nir Friedman, Michal Linial, Iftach Nachman e Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620. Citado na pág. 9

[16] **Gentles e Gallahan(2011)** Andrew J Gentles e Daniel Gallahan. Systems biology: confronting the complexity of cancer. *Cancer research*, 71(18):5961–5964. Citado na pág. 2

[17] **Griffiths** *et al.***(2008)** Anthony JF Griffiths, Susan R Wessler, Richard C Lewontin e Sean B Carroll. Introdução à genética. Em *Introdução à genética*. Guanabara Koogan. Citado na pág. 1, 5, 6

[18] **Haider e Pal(2012)** Saad Haider e Ranadip Pal. Boolean network inference from time series data incorporating prior biological knowledge. *BMC genomics*, 13(Suppl 6):S9. Citado na pág. 2

[19] **Hanahan e Weinberg(2011)** Douglas Hanahan e Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674. Citado na pág. 2

[20] **Hashimoto** *et al.***(2003)** Ronaldo F Hashimoto, Edward R Dougherty, Marcel Brun, Zheng-Zheng Zhou, Michael L Bittner e Jeffrey M Trent. Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations. *Signal Processing*, 83 (4):695–712. Citado na pág. 15

[21] **Hecker** *et al.***(2009)** Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene van Someren e Reinhard Guthke. Gene regulatoru network inference: Data integration in dynamic models-a review. *Biosystems*, 96(1):86–103. ISSN 030-2647. doi: http://dx.doi.org/10.1016/j.biosystems.2008.12.004. URL http://www.sciencedirect.com/science/article/pii/S0303264708002608. Citado na pág. 1, 9, 10

[22] **Heckerman(2008)** David Heckerman. A tutorial on learning with bayesian networks. Em DawnE. Holmes e LakhmiC. Jain, editors, *Innovations in Bayesian Networks*, volume 156 of *Studies in Computational Intelligence*, páginas 33–82. Springer Berlin Heidelberg. ISBN 978-3-540-85065-6. doi: 10.1007/978-3-540-85066-3_3. URL http://dx.doi.org/10.1007/978-3-540-85066-3_3. Citado na pág. 2, 9

[23] **Ihaka e Gentleman(1996)** Ross Ihaka e Robert Gentleman. R: a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3):299–314. Citado na pág. 25

[24] **Imoto** *et al.***(2004)** Seiya Imoto, Tomoyuki Higuchi, Takao Goto, Kousuke Tashiro, Satoru Kuhara e Satoru Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *Journal of Bioinformatics and Computational Biology*, 2(01): 77–98. Citado na pág. 2

[25] **Kanehisa** *et al.***(2002)** Minoru Kanehisa, Susumu Goto, Shuichi Kawashima e Akihiro Nakaya. The kegg databases at genomenet. *Nucleic acids research*, 30(1):42–46. Citado na pág. 2, 41

[26] **Karlebach e Shamir(2008)** Guy Karlebach e Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780. Citado na pág. 2, 9

[27] **Kauffman(1969)** Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467. Citado na pág. 1

[28] **Kelemen** *et al.***(2008)** Arpad Kelemen, Ajith Abraham e Yuehui Chen. *Computational intelligence in bioinformatics*, volume 94. Springer. Citado na pág. 9, 11

[29] **Langer-Safer** *et al.***(1982)** Pennina R Langer-Safer, Michael Levine e David C Ward. Immunological method for mapping genes on drosophila polytene chromosomes. *Proceedings of the National Academy of Sciences*, 79(14):4381–4385. Citado na pág. 8

[30] **Lareau** *et al.***(2004)** Liana F Lareau, Richard E Green, Rajiv S Bhatnagar e Steven E Brenner. The evolving roles of alternative splicing. *Current opinion in structural biology*, 14(3):273–282. Citado na pág. 7

[31] **Lee e Tzou(2009)** Wei-Po Lee e Wen-Shyong Tzou. Computational methods for discovering gene networks from expression data. *Briefings in bioinformatics*, 10(4):408–423. Citado na pág. 9

[32] **Li** *et al.***(2010)** Yong Li, Lili Liu, Xi Bai, Hua Cai, Wei Ji, Dianjing Guo e Yanming Zhu. Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks. *BMC bioinformatics*, 11(1):520. Citado na pág. 28

[33] **Lopes** *et al.***(2008)** F. M. Lopes, D. C. Martins-Jr e R. M. Cesar-Jr. Feature selection environment for genomic applications. *BMC Bioinformatics*, 9(1):451. Citado na pág. 16, 21, 22, 37

[34] **Lopes** *et al.***(2011)** F. M. Lopes, E. A. de Oliveira e R. M. Cesar-Jr. Inference of gene regulatory networks from time series by Tsallis entropies. *BMC Systems Biology*, 5(1):61. ISSN 1752-0509. doi: 10.1186/1752-0509-5-61. Citado na pág. 20, 21, 22, 37

[35] **Lopes** *et al.***(2014)** Fabricio M Lopes, Shubhra Sankar Ray, Ronaldo F Hashimoto e Roberto M Cesar. Entropic biological score: a cell cycle investigation for grns inference. *Gene*, 541(2): 129–137. Citado na pág. 2

[36] **MacQueen** *et al.***(1967)** James MacQueen *et al.* Some methods for classification and analysis of multivariate observations. Em *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, páginas 281–297. Oakland, CA, USA. Citado na pág. 28

[37] **Nakariyakul e Casasent(2009)** Songyot Nakariyakul e David P Casasent. An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, 42(9):1932–1940. Citado na pág. 12, 13, 14

[38] **Nussbaum(2008)** Robert L. Nussbaum. *Thompson e Thompson Genética Médica*. ELSEVIER MEDICINA BRASIL - TXT. ISBN 8535221492. Citado na pág. 1, 5, 6, 7, 8

[39] **Prill** *et al.***(2010)** Robert J. Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K. Sorger, Leonidas G. Alexopoulos, Xiaowei Xue, Neil D. Clarke, Gregoire Altan-Bonnet e Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PLoS ONE*, 5(2):e9202. Citado na pág. 34

[40] **Pudil** *et al.***(1994)** Pavel Pudil, Jana Novovičová e Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125. Citado na pág. 12

[41] **Sayers** *et al.***(2011)** Eric W Sayers, Tanya Barrett, Dennis A Benson, Evan Bolton, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Scott Federhen *et al.* Database resources of the national center for biotechnology information. *Nucleic acids research*, 39(suppl 1):D38–D51. Citado na pág. 2

[42] **Shalon** *et al.***(1996)** Dari Shalon, Stephen J Smith e Patrick O Brown. A dna microarray system for analyzing complex dna samples using two-color fluorescent probe hybridization. *Genome research*, 6(7):639–645. Citado na pág. 1, 8

[43] **Shannon(1949)** Claude E Shannon. Communication theory of secrecy systems. *Bell system technical*, 28(4):656–715. Citado na pág. 17, 18

[44] **Shmulevich** *et al.***(2002)** Ilya Shmulevich, Edward R Dougherty e Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792. Citado na pág. 10

[45] **Somol** *et al.***(1999)** Petr Somol, Pavel Pudil, Jana Novovičová e Pavel Paclık. Adaptive floating search methods in feature selection. *Pattern recognition letters*, 20(11):1157–1163. Citado na pág. 12

[46] **Somol** *et al.***(2004)** Petr Somol, Pavel Pudil e Josef Kittler. Fast branch & bound algorithms for optimal feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):900–912. Citado na pág. 12

[47] **Somol** *et al.***(2010)** Petr Somol, Jana Novovicová e Pavel Pudil. *Efficient feature subset selection and subset size optimization*. INTECH Open Access Publisher. Citado na pág. 12, 13

[48] **Spellman** *et al.***(1998)** Paul T Spellman, Gavin Sherlock, Michael Q Zhang, Vishwanath R Iyer, Kirk Anders, Michael B Eisen, Patrick O Brown, David Botstein e Bruce Futcher. Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297. Citado na pág. 37, 41

[49] **Stearns(1976)** Stephen D Stearns. On selecting features for pattern classifiers. Em *Proceedings of the 3rd International Joint Conference on Pattern Recognition*, páginas 71–75. Citado na pág. 12

[50] **Studham** *et al.***(2014)** Matthew E Studham, Andreas Tjärnberg, Torbjörn EM Nordling, Sven Nelander e Erik LL Sonnhammer. Functional association networks as priors for gene regulatory network inference. *Bioinformatics*, 30(12):i130–i138. Citado na pág. 2

[51] **Studio(2012)** R Studio. R studio: integrated development environment for r, 2012. Citado na pág. 26

[52] **Szklarczyk** *et al.***(2011)** D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen e C. von Mering. http://string-db.org/newstring_cgi/show_input_page.pl?UserId=t_SleQHMkmBC&sessionId=JwQPmOKggW7c, 2011. [Online; accessed 11-Maio-2015]. Citado na pág. 26

[53] **Szklarczyk** *et al.***(2011)** Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Minguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork *et al.* The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic acids research*, 39(suppl 1):D561–D568. Citado na pág. 2, 25

[54] **Szklarczyk** *et al.***(2015)** Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P. Tsafou, Michael Kuhn, Peer Bork, Lars J. Jensen e Christian von Mering. STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 43(Database issue):D447–D452. Citado na pág. 26

[55] **Theodoridis e Koutroumbas(2001)** Sergios Theodoridis e Konstantinos Koutroumbas. Pattern recognition and neural networks. Em *Machine Learning and Its Applications*, páginas 169–195. Springer. Citado na pág. 12, 13

[56] **Tomotani(2010)** Barbara Mizumo Tomotani. Aspectos evolutivos do splicing alternativo. *REVISTA DA BIOLOGIA, www.ib.usp.br/revista*, 4. Citado na pág. 7

[57] **Tsallis(1988)** Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *of Statistical Physics*, 52(1-2):479–487. Citado na pág. 17, 20

[58] **Tsallis** *et al.***(2005)** Constantino Tsallis, Murray Gell-Mann e Yuzuru Sato. Extensivity and entropy production. *Europhysics News*, 36(6):186–189. Citado na pág. 21

[59] **Velculescu** *et al.***(1995)** Victor E Velculescu, Lin Zhang, Bert Vogelstein e Kenneth W Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484. Citado na pág. 1, 8

[60] **von Mering** *et al.***(2005)** Christian von Mering, Lars J. Jensen, Berend Snel, Sean D. Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A. Huynen e Peer Bork. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(Database issue). Citado na pág. 26

[61] **Wang** *et al.***(2009)** Zhong Wang, Mark Gerstein e Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63. Citado na pág. 1, 8

[62] **Warde-Farley** *et al.***(2010)** David Warde-Farley, Sylva L Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes *et al.* The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl 2):W214–W220. Citado na pág. 2

[63] **Webb(2002)** A. R. Webb. *Statistical Pattern Recognition*. John Willey & Sons, 2nd edição. Citado na pág. 33

[64] **White e Salamonsen(2005)** Christine A White e Lois A Salamonsen. A guide to issues in microarray analysis: application to endometrial biology. *Reproduction*, 130(1):1–13. Citado na pág. 8, 9

[65] **Zhao** *et al.***(2008)** Wentao Zhao, Erchin Serpedin e Edward R Dougherty. Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 5(2):262–274. Citado na pág. 17