**Universidade Tecnológica Federal do Paraná**
**Programa de Pós-Graduação Engenharia Elétrica e Informática Industrial**

**Leandro Takeshi Hattori**

# INFERÊNCIA DE REDES DE REGULAÇÃO GÊNICA UTILIZANDO MÉTODOS DE BUSCA E OTIMIZAÇÃO

**Dissertação – Mestrado**

**Curitiba**
**2016**

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

LEANDRO TAKESHI HATTORI

# INFERÊNCIA DE REDES DE REGULAÇÃO GÊNICA UTILIZANDO MÉTODOS DE BUSCA E OTIMIZAÇÃO

DISSERTAÇÃO

**CURITIBA**

**2016**

LEANDRO TAKESHI HATTORI

# INFERÊNCIA DE REDES DE REGULAÇÃO GÊNICA UTILIZANDO MÉTODOS DE BUSCA E OTIMIZAÇÃO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Mestre em Ciências" – Área de Concentração: Engenharia da Computação.

Orientador: Heitor Silvério Lopes

Co-orientador: Fabrício Martins Lopes

**CURITIBA**

**2016**

**Dados Internacionais de Catalogação na Publicação**

H366i 2016

Hattori, Leandro Takeshi
Inferência de redes de regulação gênica utilizando métodos de busca e otimização / Leandro Takeshi Hattori.-- 2016.
75 f.: il.; 30 cm

Texto em português, com resumo em inglês.
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2016.
Bibliografia: f. 69-75.

1. Computação inspirada biologicamente. 2. Bioinformática. 3. Regulação genética - Modelos matemáticos. 4. Transcrição genética - Regulamento. 5. Redes complexas. 6. Algoritmos genéticos. 7. Engenharia elétrica - Dissertações. I. Lopes, Heitor Silvério, orient. II. Lopes, Fabrício Martins, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD: Ed. 22 -- 621.3

**Biblioteca Central da UTFPR, Câmpus Curitiba**

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
*Câmpus Curitiba*

*Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial*

**Título da Dissertação Nº. ____**

# Inferência De Redes De Regulação Gênica Utilizando Métodos De Busca E Otimização.

por


## Leandro Takeshi Hattori

**Orientador: Prof. Dr. Heitor Silvério Lopes (UTFPR)**
**Coorientador: Prof. Dr. Fabrício Martins Lopes (UTFPR)**


Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: **Engenharia de Computação** do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14 **h** do dia 22 de março 2016. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

Prof. Dr. Fabrício Martins Lopes
(Presidente – UTFPR)

Prof. Dr. David Corrêa Martins Junior
(UFABC)

Prof. Dr. César Manuel Vargas Benítez
(UTFPR)

Visto da coordenação:

Prof. Dr. Emílio Carlos Gomes Wille
(Coordenador do CPGEI)

## AGRADECIMENTOS

Ao meu orientador e co-orientador Heitor Silvério Lopes e Fabrício Martins Lopes, pelas valiosas orientações neste trabalho; por suas amizades com as conversas e discussões que levarei como lições para o resto da minha vida; suas dedicações à pesquisa, que sempre me motivaram a alcaçar meus objetivos como pesquisador;

Ao grupo do BIOINFO-IC: Ademir Cristiano Gabardo, André Macário Barros, César Manuel Benítez Vargas, Fernando Carvalho de Souza, Glaucio Porcides Czekailo, Hugo Alberto Perlin, Jonas Krause, Lia Ayumi Takiguchi, Manassés Ribeiro, Rodrigo Silva pelo companheirismo nos bons e nos piores momentos;

Agradeço especialmente aos Professores Fabrício Martins Lopes e Henrique Yoshikazu Shishido por terem acreditado e confiado em mim desde os tempos da Iniciação Científica;

Agradeço aos Professores do CPGEI e UTFPR-CP pelos valiosos conhecimentos e experiências compartilhadas.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela bolsa de estudos concedida;

Às famílias Michelato e Yoshiy sempre presentes como uma família para mim, a todo o momento me apoiando e me recebendo de braços abertos;

E, a todos que de alguma forma contribuíram para a realização desse trabalho e conclusão desse período tão especial em minha vida.

Obrigado!

# RESUMO

HATTORI, T. Leandro. INFERÊNCIA DE REDES DE REGULAÇÃO GÊNICA UTILIZANDO MÉTODOS DE BUSCA E OTIMIZAÇÃO. 78 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Para melhor entender os mecanismos de controle celular, várias abordagens tem sido desenvolvidas para inferir Redes de Regulação Gênica (GRNs) utilizando dados temporais de expressão gênica. Entretanto, a grande quantidade de genes observados em contraste com as poucas amostras de expressão gênica disponíveis torna a inferência de GRNs um dos problemas mais importantes na Bioinformática. Nesta dissertação o problema de inferência de GRNs é decomposto em *n* subproblemas de seleção de características. Para cada subproblema são obtidos os genes preditores para cada gene alvo. O método de seleção de característica é basicamente composto por um algoritmo de busca e a função critério. Neste trabalho foram utilizados algoritmos bioinspirados (ED, AM e CAA) e de busca sequencial (BSF e BSFF), como função critério foi utilizada a Entropia Condicional Média (ECM). Também foram propostos métodos de pós-processamento para a otimização da GRN inferida pelos algoritmos bioinspirados: o algoritmo de Quine-McCluskey (QM) e uma rede de consenso gerada a partir das redes inferidas pelos algoritmos bioinspirados com a otimização do algoritmo de QM. Para os experimentos de inferência foram exploradas Redes Artificiais Gênicas (AGNs) baseadas em Redes Booleanas Probabilísticas (RBPs), variando características de topologia, média de ligações, número de genes e quantidade de dados de expressão gênica. Os resultados mostraram que o algoritmo ED obteve melhores resultados de precisão quando comparado com os algoritmos sequenciais. Quando comparado com outros algoritmos bioinspirados, o ED também obteve melhores resultados do que o AM e CAA. No experimento de otimização das redes inferidas pelos algoritmos bioinspirados, o algoritmo de QM apresentou um bom desempenho, removendo genes preditores que não estavam contidos na rede real, levando a uma melhora na precisão da rede inferida e sua similaridade com a rede real. A rede de consenso apresentou resultados de precisão e similaridade melhores do que aqueles obtidos pelos métodos bioinspirados somente. Os resultados alcançados sugerem que a aplicação do método de consenso dos algoritmos bioinspirados com a otimização de QM é bastante promissor para o problema de inferência de GRNs.

**Palavras-chave:** Computação Bioinspirada, Busca Sequencial, Bioinformática, Seleção de Características, Rede de Regulação Gênica, Redes Complexas, Quine-McCluskey

# ABSTRACT

HATTORI, T. Leandro. INFERENCE OF GENE REGULATORY NETWORS USING OPTIMIZATION AND SEARCH METHODS. 78 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

For better understanding the mechanics of cellular control, many different approaches have been developed for inferring Gene Regulatory Networks (GRNs), using temporal gene expression data. However, the large amount of genes observed in contrast with the small amount of gene expression samples makes the inference of GRNs one of the most important problems in Bioinformatics. In this dissertation, the inference of GRNs is a problem decomposed in $n$ feature selection sub-problems. For each sub-problem, the predictor genes for each target gene are obtained. Basically, the feature selection method is composed by a search algorithm and a criterion function. In this work we used bioinspired methods (DE, BAT and ABC) and sequential search methods (SFS and SFFS), and the criterion function we used the Mean Conditional Entropy (MCE). Also, we proposed some pos-processing methods for the optimization of the GRN inferred by the bioinspired methods, by using the Quine-McCluskey (QM) algorithm as well as a consensus network generated from the networks inferred by the bioinspired methods and later optimized by the QM. For the inference experiments, we explored Artificial Genic Networks (AGN) based on Probabilistic Boolean Networks (PBNs), changing the features of the topology, the average number of connections, the number of genes, and the amount of gene expression data. Results showed that the DE algorithm obtained better results, regarding accuracy, when compared with the sequential search methods. When compared with the other bioinspired methods, DE also achieved better results than BAT and ABC. For the optimization of the inferred networks by the bioinspired methods, the QM algorithm presented a good performance, removing predictor genes that were not contained in the real network, leading to an improvement of the accuracy of the inferred network, and its similarity with the real one. The consensus network presented accuracy results and similarity even better than those obtained by the bioinspired methods alone. Overall results suggest that the application of the consensus method based on the bioinspired methods together with the QM pos-processing is promising for the GRNs inference problem.

**Keywords:** Bio-inspired Computation, Sequential Search, Bioinformatics, Feature Selection, Gene Regulatory Network, Complex Network, Quine-McCluskey

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE SIGLAS

| | |
|---|---|
| AG | Algoritmo Genético |
| AGN | *Artificial Gene Network* |
| ARACNE | *Algorithm for the Reconstruction of Accurate Cellular NEtworks* |
| BA | Barabási-Albert |
| BG | Boltzmann-Gibbs |
| BSFF | Busca Sequencial Flutuante para Frente |
| BSF | Busca Sequencial para Frente |
| BST | Busca Sequencial para Trás |
| CAA | Colônia Artificial de Abelhas |
| cDNA | *Complementary DNA* |
| CE | Computação Evolucionária |
| CIA | Critério de Informação de Akaike |
| CLR | *Context Likehood of Relatedness* |
| DNA | *Deoxyribonucleic Acid* |
| DREAM | *Dialogue on Reverse Engineering Assessment and Methods* |
| ECM | Entropia Condicional Média |
| EDD | Evolução Diferencial Discretizado |
| ED | Evolução Diferencial |
| EDOEP | Evolução Diferencial Otimização por Enxame de Partículas |
| EE | Estratégia Evolucionária |
| ER | Erdös-Rényi |
| FN | Falso Negativo |
| FP | Falso Positivo |
| GRN | *Gene Regulatory Network* |
| IE | Inteligência de Enxames |
| IM | Informação Múltua |
| MK | Mapa de Karnaugh |
| ncRNA | *noncoding RNA* |
| OCF | Otimização de Colônia de Formigas |
| OEP | Otimização de Enxame de Partículas |
| pECO | *parallel ECOlogically-inspired* |
| PE | Programação Evolucionária |
| PG | Programação Genética |
| PIM | Predição Intrinsecamente Multivariada |
| RA | Regras de Associação |
| REVEAL | *REVerse Engineering ALgorithm* |
| RNAm | *RiboNucleic Acid messenger* |
| TF | Tabela de Frequência |
| VN | Verdadeiro Negativo |
| VP | Verdadeiro Positivo |
| WS | Watts-Strogatz |

## LISTA DE SÍMBOLOS

| | |
|---|---|
| $\langle k \rangle$ | Média de ligações entre os genes da rede |
| $y$ | Gene alvo |
| $\xi$ | Saída *don't care* na tabela verdade |
| X | Conjunto dos genes da rede |
| E | Conjunto das arestas da rede |
| $\psi$ | Conjunto das função de transição |
| $\vec{\chi}$ | Individuo da população |
| $\vec{V}^G$ | Vetor doador |
| $\vec{U}_i^G$ | Vetor teste |
| $CR$ | Taxa de *crossover* |
| $r$ | Taxa de pulso |
| $A$ | Amplitude |
| H(X) | Entropia do conjunto X |
| $H(Y \mid x)$ | Entropia condicional de Y dado que as instâncias de x foi observada |
| $H(Y \mid X)$ | Entropia condicional de Y dado que as instâncias do conjunto X foi observada |
| $G$ | Representa um grafo ou uma rede |
| $E$ | Conjunto das arestas da rede |
| $S$ | Conjunto dos estados que uma rede pode assumir |
| $M$ | Matriz de adjacência |
| $\vec{s}_t$ | Estados dos genes da rede no instante t |

**SUMÁRIO**

# 1 INTRODUÇÃO

Um organismo vivo pode ser estudado como uma rede de interação entre moléculas de diferentes níveis, tais como nível genômico, o nível transcriptômico e nível proteômico. A Rede de Regulação Gênica (GRN, do inglês *Gene Regulatory Network*) é um tipo de rede que é capaz de apresentar uma visão geral sobre as interações entre os diferentes níveis do sistema biológico. A partir da descoberta de uma GRN é possível responder algumas perguntas: como um determinado sistema biológico pode responder a diversos estímulos externos, quais são as mudanças no estado do sistema sob certas condições e quais são as alterações na rede, caso algum elemento do sistema biológico esteja anormal. Dados as tais possibilidades, a descoberta e utilização destas redes vem se destacando, sendo assim amplamente utilizadas para análise de doenças, como o mal de Alzheimer (ZHANG et al., 2015) e o Câncer (MADHAMSHETTIWAR et al., 2012; SIMÕES et al., 2015), desenvolvimento de novas drogas, conhecimento geral sobre os mecanismos de controle, entre outras aplicações.

A inferência de GRNs a partir de dados temporais de expressão gênica é baseada no Dogma Central da Biologia Molecular (CRICK et al., 1970), o qual postula que a expressão gênica tem uma forte influência sobre o sistema de um organismo. Assim como no processo de inferência de GRNs, diversos outros tipos de problemas também utilizam os dados de expressão gênica, como para a análises de tecidos tumorais contra amostras de controle, estudar o perfil de expressão temporal de anomalias, identificação de genes importantes, identificação de genes expressos em cada etapa do desenvolvimento de um organismo, entre outros casos. Dada a esta demanda, a tecnologia de mensuração de dados de expressão gênica está sendo alvo de um contínuo desenvolvimento, principalmente nos aspectos da quantidade de genes que podem ser analisados simultaneamente e da precisão do nível de expressão. A tecnologia mais recente e mais precisa para análise de expressão é o RNA-Seq (WANG et al., 2009), entretanto o *DNA microarray* (VELCULESCU et al., 1995) ainda é a tecnologia mais difundida, em razão do baixo custo do experimento.

Um cenário bastante comum para recuperar a estrutura das GRNs é dispor de poucos experimentos de expressão gênica (amostra de treinamento), em contraste com milhares de

genes avaliados (espaço de características). Este fenômeno é chamado de "maldição da dimensionalidade", no qual o espaço de características é muito grande e existem poucas amostras para poder concentrar os dados de uma determinada classe (BISHOP, 1995). Uma solução comumente utilizada nesta área são as Redes Artificiais Gênicas (AGN, do inglês *Gene Regulatory Network*), que permitem simular dados temporais de expressão gênica utilizando uma rede de interação gênica *in silico* baseadas em topologias de redes complexas.

Atualmente é difícil encontrar dados de expressão gênica obtidos a partir de uma grande quantidade de experimentos, e igualmente difícil encontrar redes de regulação gênica que possuam informações sobre todas as ligações entre os genes de um organismo e com uma quantidade de experimentos de expressão temporal satisfatória. Dada a estas dificuldades, as AGNs também são um meio satisfatório de obter a estrutura da rede e dados de expressão. Outro recurso para este tipo de problema é procurar na literatura as GRNs *gold standard*, as quais fornecem os dados de expressão e estrutura da rede.

Quando se observa o comportamento de um gene dependente de uma função dada a expressão de um conjunto de genes, diversas possibilidades podem ser consideradas para identificar as relações entre causa-efeito. Este problema pode ser identificado como um problema de seleção de características, em que a expressão do gene alvo pode ser altamente definida por um conjunto de genes preditores. O desafio é encontrar os genes preditores (características), no conjunto total de genes, que possuem grandes probabilidades de serem genes que possuem influência sobre o gene alvo. Entretanto, encontrar os melhores preditores para cada gene alvo da rede não é uma tarefa trivial. Chickering et al. (2004) mostrou que a complexidade do problema para encontrar a melhor estrutura da rede em grandes espaços de características é *NP-hard*. Neste contexto diversas abordagens foram propostas na literatura (LOPES et al., 2008; MARBACH et al., 2012; JIMENEZ et al., 2015).

Comumente, o método de seleção de características incluem dois elementos: a função critério para avaliar a qualidade dos subconjuntos de características (genes preditores) e um algoritmo de busca, que percorre o espaço de características combinando características para encontrar o melhor subconjunto. Dada a exponencial complexidade da seleção de características meta-heurísticas apresentam bom custo-benefício, comparando o custo computacional pela qualidade da solução, quando o tamanho do problema aumenta. Neste sentido a aplicação de métodos de busca sequencial e de algoritmos bioinspirados para a inferência de GRNs e o uso de métodos de otimização das GRNs inferidas são as principais motivações deste trabalho.

## 1.1 OBJETIVO

### 1.1.1 OBJETIVO GERAL

O objetivo deste trabalho é a investigação de métodos computacionais de inferência de GRNs, focando no aspecto de identificação (inferência) a partir de dados temporais de expressão gênica.

### 1.1.2 OBJETIVOS ESPECÍFICOS

- Utilizar os algoritmos de buscas sequenciais Busca Sequencial para Frente (BSF) e Busca Sequencial Flutuante para Frente (BSFF) para encontrar os subconjuntos de genes preditores;
- Utilizar os algoritmos bioinspirados Evolução Diferencial (ED), Colônia Artificial de Abelha (CAA) e Algoritmo do Morcego (AM) para encontrar os subconjuntos de genes preditores;
- Comparar resultados encontrados pelos algoritmos BSF e BSFF e o algoritmo ED;
- Comparar resultados dos algoritmos bioinspirados de Inteligência de Enxames (IE) AM e CAA e de Computação Evolucionária (CE) ED;
- Utilizar métodos de otimização da rede inferidas tais como: Quine-McCluskey (QM) e Rede de Consenso;

## 1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho é organizado da seguinte forma: No Capítulo 2 são apresentados os fundamentos teóricos relacionados ao trabalho. Os métodos são apresentados no Capítulo 3 e os resultados obtidos a partir da metodologia são apresentados no Capítulo 4. Por fim, no Capítulo 5 são apresentadas as conclusões do trabalho e os trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1 BIOLOGIA MOLECULAR

O Ácido DesoxirriboNucleico (DNA, *DeoxyriboNnucleic Acid*) é uma estrutura que armazena toda a informação biológica de um organismo, este material é formado por duas fitas de nucleotídeos, onde cada nucleotídeo de uma fita é pareado com o nucleotídeo complementar da outra fita. Cada informação no DNA está transcrita em uma sequência de nucleotídeos, denominada gene. A expressão (ativação) do gene é o início do processo da decodificação da informação armazenada para a obtenção do produto biológico, como proteínas, enzimas e RNAs. A expressão gênica é dada a partir de fatores externos ou internos do organismo (VOET et al., 2008).

O primeiro modelo que descreve o fluxo de expressão gênica para a síntese proteica foi proposto por Francis Cricks em 1958 (CRICK, 1958), chamado de Dogma Central da Biologia Molecular ou ”Hipótese da Sequência”. Este modelo leva em consideração apenas o fluxo em cascata da transcrição do gene para um RNA e posteriormente para a síntese de uma proteína. No decorrer dos estudos outros fluxos foram descobertos e integrados à complexidade do modelo, como a transcrição reversa, onde ocorre a integração do RNA no DNA e a autorreplicação de RNA, como apresentado na Figura 1. Também foi descoberto que a maior parte dos genes não codifica uma proteína, estes genes transcrevem RNAs não codificantes (ncRNA, *Non Colding RNA*), os quais possuem funções reguladoras pós-transcricionais (DINGER et al., 2008).

A seguir serão apresentados os principais conceitos e características sobre os dados de expressão gênica e redes de regulações gênicas.

### 2.1.1 DADOS DE EXPRESSÃO GÊNICA

Muitos problemas na área de análise genômica dependem do conhecimento de quais são as sequencias de DNA e RNA e qual a abundância do produto da expressão em diversos tipos

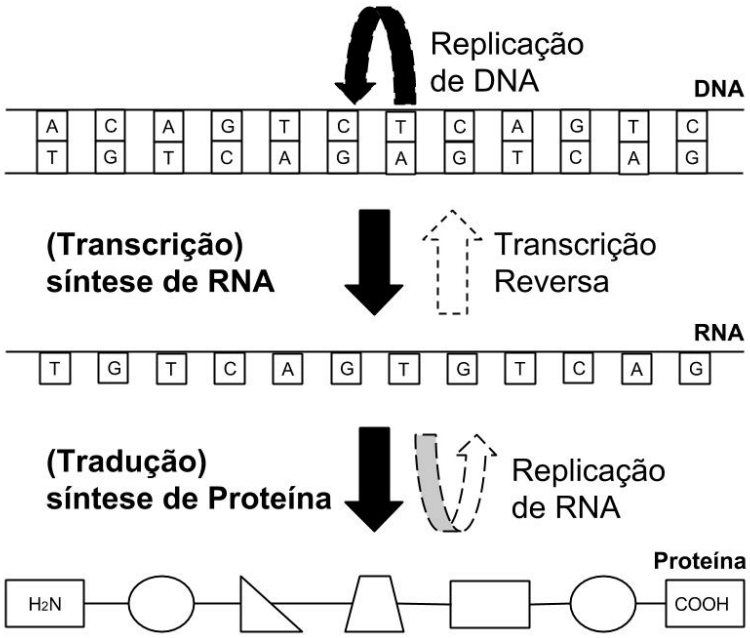



**Figura 1: Fluxo do Dogma Central da Biologia Molecular Atualizado. As setas em preto representam o primeiro modelo de transcrição. As setas em branco representam os fluxos adicionados ao modelo.**

**Fonte: Adaptado de (LEHNINGER, 1989)**

de situações. Compreender quais são as diferenças do nível de expressão entre diferentes tipos de genes, tecidos, situações, organismos pode trazer importantes informações, e consequentemente aprender como o sistema de um organismo funciona. Desta forma é possível estudar um indivíduo no nível transcriptômico e entender detalhadamente como doenças e anomalias genéticas se comportam comparando com organismos controle.

Uma abordagem comumente utilizada para o reconhecimento de padrões nestes tipos de problemas é a utilização da tecnologia de DNA *microarrays* ou bio-chips de DNA (SHALON et al., 1996). Os DNA *microarrays* permitem avaliar os níveis de expressão em grandes escalas, quantificando o nível de expressão pela quantidade de Ácido Ribonucleico mensageiro (RNAm, *RiboNucleic Acid messenger*).

O DNA *microarray* é uma superfície sólida (*array*) com diversos pontos microscópicos (*spots*), nos quais são depositadas quantidades precisas de DNA em cada *spot*, formando uma matriz de microarranjo (*microarray*). Para obter o nível de expressão de um determinado gene é feito primeiramente isolamento da amostra de interesse. Posteriormente, são adicionados cDNAs (cDNA, *Complementary DNA*) específicos que se ligam aos RNAms da amostra. Estes cDNAs funcionam como componentes fluorescentes que, quando excitados, geram luminescência,

como mostrado na Figura 2. A seguir a amostra com cDNAs passa pelo processo de lavagem, que remove os cDNAs que não se ligaram ou que restaram da hibridização. Na sequencia, esta amostra é incubada para ser escaneada para a digitalização da imagem. Existem dois tipos de *scanners*: o *Scanner* CCD, que excita as lâminas com uma luz branca e uma câmera fotografa a emissão gerada a partir dos componentes fluorescentes presentes na amostra (ESTEVES, 2002) e o *Scanner* a *laser*, que varre as laminas incidindo a amostra com raio *laser* de diversos comprimentos de ondas específicos digitalizando a imagem. Por fim, o nível de luminosidafexcide da amostra de cada *spot* é digitalizada e convertida em um valor numérico.

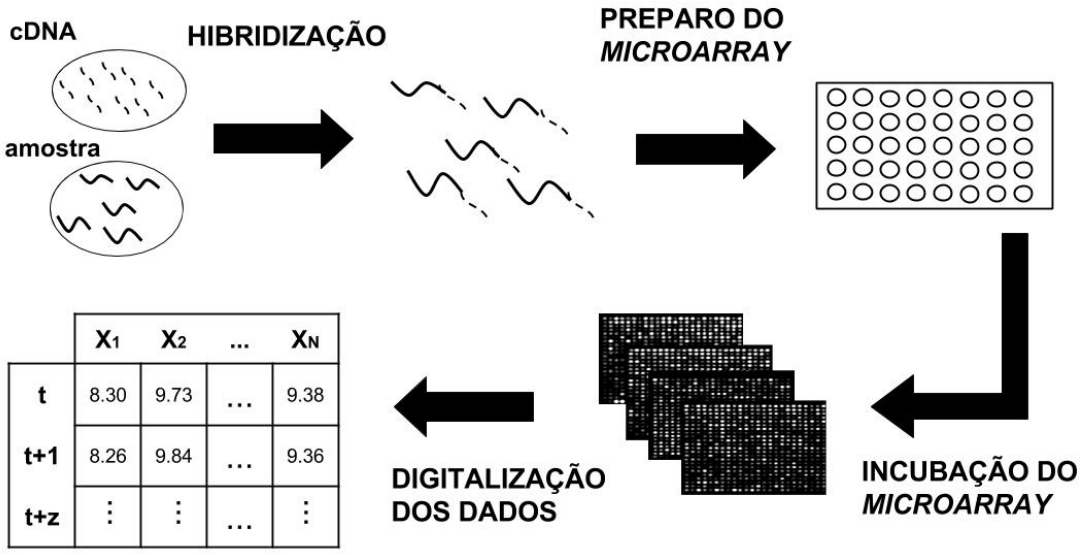



**Figura 2: Processo geral para obter o nível de expressão gênica a partir de uma amostra biológica. Fonte: Adaptado de (LOPES, 2011).**

## 2.2 REDES COMPLEXAS

O estudo da teoria dos grafos surgiu a partir do trabalho de Leonard Euler que propôs a solução do problema das pontes de Konigsberg em 1736. O problema tratado era como passar pelas sete pontes de Konigsberg apenas uma única vez. Para a solução do problema os componentes foram representados através de uma rede, onde as pontes representavam as arestas e as regiões pelas quais as pontes se interligavam foram representadas por vértices.

Em um primeiro instante, pensava-se que as redes reais poderiam ser representadas por modelos de redes aleatórias, conforme o modelo, proposto por primeiramente Paul Erdos e Alfred Renyi em 1959 (ERDÖS; RÉNYI, 1959), e que utilizava relacionamentos entre os vértices por padrões aleatórios, tal qual apresentado na Figura 3. Posteriormente, notou-se que

redes aleatórias eram incapazes de representar a natureza de sistemas reais. Então, outros modelos foram propostos como o modelo do mundo pequeno (*small world*) (WATTS; STROGATZ, 1998), livre escala (*scale free*) (*scale free*) (BARABÁSI; ALBERT, 1999), entre outros.

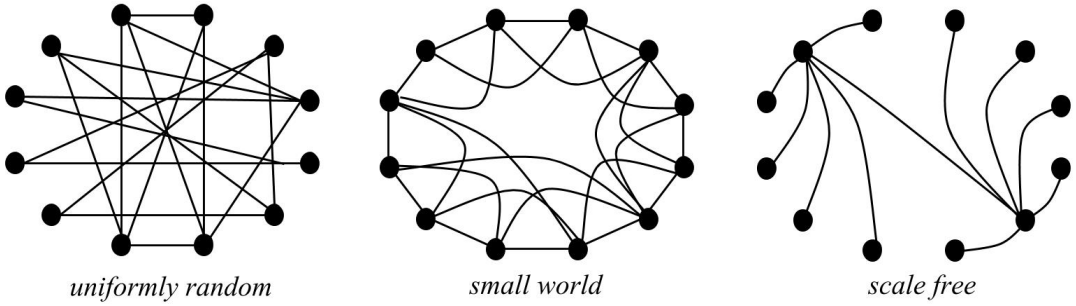



**Figura 3: Topologias de redes complexas: *uniformly random* (ER, Erdös-Rényi), *small world* (WS, Watts-Strogatz) e *scale free* (BA, Barabási-Albert).**

**Fonte: Adaptado de (HUANG et al., 2005).**

Considerando as características de flexibilidade e generalidade das redes complexas é possível adicionar medidas e métodos estabelecidos a partir de um sistema real (COSTA et al., 2007). Utilizando-se destas características nos dados extraídos de um sistema real, o objetivo da rede é a representação das interações entre os elementos da rede. Quando são considerados dados temporais é possível tratar a rede de forma dinâmica, ou seja, sua estrutura pode ser modificada ao longo do tempo.

As redes complexas foram propostas como meio de representação da estrutura de interesse para diversos tipos de problemas, inclusive para representação de sistemas biológicos. Uma destas representações biológicas é a representação de uma GRN a partir de redes complexas (LOPES et al., 2011, 2008; TERFVE et al., 2012), as topologias de redes complexas e as suas propriedades intrínsecas podem representar adequadamente GRNs.

Uma rede $G = (X, E)$ pode ser representada por meio de um conjunto de $X = \{x_1, .., x_N\}$ de vértices e por um conjunto de $E = \{e_1, e_2, \ldots, e_M\}$ de arestas. Uma rede é definida pela quantidade de $N$ vértices e por uma quantidade de $M$ arestas conectada aos vértices com um grau médio $\langle k \rangle$ de conexões para cada vértice da rede.

A seguir serão apresentados os principais modelos de redes complexas bem como suas respectivas características.

### 2.2.1 MODELO DE ERDÖS E RÉNYI

O modelo desenvolvido pelos matemáticos ERDÖS e RÉNYI (1959) (ER) é o modelo de redes complexas mais simples, e se baseia no seguinte processo. Basicamente, a partir de uma rede de $n$ vértices desconectados são inseridas $M$ ligações entre os vértices que são distribuídas com probabilidades $P$ uniformes. Entretanto, evita-se os auto relacionamentos e as conexões múltiplas nos vértices da rede.

No modelo ER a média de ligações $\langle k \rangle$ dos $n$ vértices é definido por $\langle k \rangle = P(n-1)$. A probabilidade de um vértice $x_i$ se ligar a outro vértice $x_j$ é dada pela Equação 1.

$$P(x_i, x_j) = \frac{\langle k \rangle}{n-1} \tag{1}$$

Considerando a distribuição de Poisson com relação às conexões entre os vértices do modelo ER (COSTA et al., 2007), apresentada pela Equação 2, esta rede também pode ser denominada de Poisson *random graphics*.

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!} \tag{2}$$

### 2.2.2 MODELO DE WATTS E STROGATZ

O modelo proposto por Watts e Strogatz (1998)(WS) é uma abordagem que contrasta com o modelo ER. Ou seja, este modelo parte do pressuposto que nem todos os sistemas possuem características totalmente aleatórias. O nome dado a este modelo é em razão do fenômeno denominado mundo pequeno (*small-world*), apresentado pelo pesquisador Stanley Milgran (MILGRAM, 1967), que mostrou que a distância entre quaisquer pessoas nos Estados Unidos era de aproximadamente seis. No modelo WS este fenômeno ocorre de forma similar, onde a partir de um vértice da rede, na maioria das vezes, é possível alcançar qualquer outro vértice percorrendo um baixo número de nós.

No modelo WS os $n$ vértices da rede são ligados em uma topologia de anel, onde cada vértice possui $k$ arestas que são ligadas aos seus vizinhos mais próximos. A seguir, as $k$ arestas dos $n$ vértices tem uma probabilidade $p$ de serem reconectadas de forma aleatória. Assim, é possível produzir uma rede intermediária ($0 < p < 1$), sendo que $p = 0$ gera uma rede regular e $p = 1$ gera uma rede aleatória.

Existem duas propriedades que caracterizam uma rede *small-world*: o tamanho do caminho $L(p)$ mais curto entre dois vértices e o coeficiente de agrupamento $C(p)$, que mede o grau de conectividade dos vértices. Ambas as redes ER e WS possuem a propriedade apresentada

por Milgran, entretanto apenas as redes WS possuem um alto $C(p)$.

### 2.2.3 MODELO DE BARABÁSI E ALBERT

Nas topologias WS e ER cada vértice da rede possui uma média de conexões. Por outro lado, na topologia de BA a distribuição das conexões dos vértices é desproporcional, existindo poucos vértices muito conectados e muitos vértices poucos conectados.

Estudos na dinâmica e na estabilidade de sistemas reais mostraram que a probabilidade $P(k)$ de um vértice da rede interagir com $k$ outros vértices decai com uma lei de potência, de acordo com a Equação 3.

$$P(k) \sim k^{-\gamma} \tag{3}$$

onde $\gamma$ representa a constante de decaimento exponencial.

Para estabelecer as conexões de uma rede baseada na topologia de BA são utilizadas duas regras: o crescimento, que a cada passo de tempo é adicionado um novo nó com $\langle k \rangle$ arestas conectadas a nós presentes no sistema. E a regra de conexão preferencial entre os nodos, no qual assume que um novo nó será conectado ao nó $i$ de acordo com sua conectividade.

A probabilidade de conexão de um vértice $x_j$ a um vértice $x_i$ é definida a partir do grau de conexão do vértice $x_j$, ou seja, depende linearmente ao grau $k_j$, como mostrado na Equação 4.

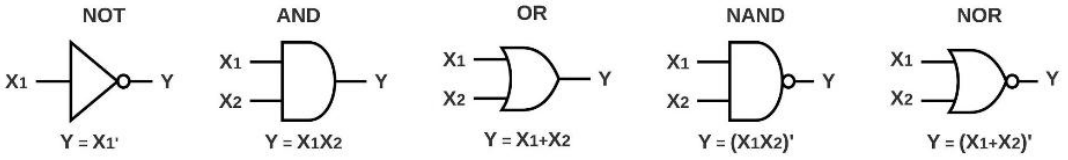
$$P(x_i, x_j) = \frac{k_j}{\sum_u k_u}, \forall x_u \in X. \tag{4}$$

## 2.3 CIRCUITOS DIGITAIS

Os circuitos são divididos em duas abordagens, os circuitos analógicos, que trabalham com valores contínuos, e os circuitos digitais, que trabalham com valores discretos. Circuitos digitais operam apenas com um número finito de estados, como os circuitos digitais de estados binários, que possuem apenas dois estados, 0 e 1. A principal vantagem de se trabalhar com circuitos digitais são: maior tolerância a ruídos, podendo ser tanto armazenados e duplicados sem perda de informação. Possuem maior eficiência para manipulação e existem diversos métodos tanto para a detecção quanto correção de possíveis erros.

### 2.3.1 PORTAS LÓGICAS

As portas lógicas são os elementos primários para a criação de um circuito lógico e podem ser representadas com símbolos ou expressões Booleanas, como mostrado na Figura 4. As principais portas lógicas são as portas *AND*, *OR* e *NOT*, e com tais portas é possível gerar qualquer circuito lógico. Na Figura 4, com exceção da porta *NOT*, que possui apenas uma entrada, os outros operadores possuem aridade igual a dois. Outro fator muito importante é que qualquer porta pode ser combinada com qualquer outra porta, permitindo uma infinidade de processos que podem ser aplicados aos sinais digitais.




| X1 | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

,

| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

,

| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

,

| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

,

| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Figura 4: Símbolos dos operadores lógicos *NOT*, *OR*, *AND*, *NAND* e *NOR*, suas respectivas representações em forma de expressão e tabelas verdades.**

**Fonte: Adaptado de (TOCCI et al., 2003).**

Cada operador Booleano possui uma característica, no qual dada uma entrada existe uma saída, assim como no operador *NOT* (negação) a saída do valor é o inverso da entrada, ou seja, caso a entrada seja igual a 1 a saída 0 e vice versa. Considerando o operador *AND* (E) é caracteriza do pela saída igual a 1 apenas quando ambas as entradas forem igual a 1. O operador *OR* (OU) fornece uma saída igual a 1 quando pelo menos uma das entradas for igual a 1. A porta lógica *NAND* (NÃO E) é uma combinação das portas *AND* e *NOT*, ou seja, a saída é o inverso da saída de uma porta *AND*. A porta lógica *NOR* (NÃO OU) é a combinação das portas *OR* e *NOT* e também possui a saída invertida da porta *OR*. Existem também outras portas lógicas como a *XOR* (OU exclusivo) que pode ser constituída por duas portas *AND*, e duas portas *NOT* e uma porta *OR*.

### 2.3.2 TABELA-VERDADE

A tabela-verdade é um importante recurso para projetos de circuitos digitais, esta tabela representa as saídas de um circuito lógico com base nas entradas. Apesar do exemplo mostrado na Tabela 1 possuir apenas 3 variáveis ($x_1$, $x_2$ e $x_3$), a tabela-verdade pode receber quantas variáveis forem necessárias no circuito. As linhas da tabela verdade das variáveis de entrada representam as combinações dos possíveis estados binários e suas respectivas saídas, apresentadas na coluna $y$. A variável $y$ pode assumir tanto valores 0, 1 e $\xi$. O símbolo $\xi$ representa que para as entradas $x_i$ a saída $y$ não é importante (TOCCI et al., 2003).

**Tabela 1: Exemplo de uma tabela verdade de um circuito digital com três entradas ($x_1$, $x_2$ e $x_3$) e uma saída $y$. Os estados que as entradas podem assumir são 0 e 1, enquanto a saída pode assumir 0, 1 e $\xi$.**

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | $\xi$ |

**Fonte: Autoria própria.**

### 2.3.3 MÉTODOS DE SIMPLIFICAÇÃO

#### 2.3.3.1 MAPA DE KARNAUGH

O Mapa de Karnaugh (MK) permite simplificar um circuito digital (FLOYD, 2007). Dada a complexidade de implementação com mais de 6 variáveis, comumente são encontradas aplicações com restrições de até 6 variáveis para solucionar MK. No MK os valores de $x_i = 0$ e $x_i = 1$ são representados respectivamente por $\overline{x_i}$ e $x_i$ para a soma de produtos. Na Figura 6 são apresentados exemplos do processo de conversão de um tabela verdade com duas e três variáveis para uma expressão Booleana. No Exemplo 6(a) as variáveis $x_1 = 0$, $x_2 = 0$ e $y = 1$ da tabela é representado no MK no quadrado $\overline{x_1}\ \overline{x_2}$ com valor igual a $y$, ou seja, igual a 1. Utilizando o mesmo procedimento os valores da tabela verdade são mapeados para o MK. Um fator muito importante é a ordem das posições das variáveis que apresentadas na horizontal e vertical do MK devem respeitar a menor distância de Hamming (TOCCI et al., 2003). A distância de Hamming entre duas *strings*, de mesmo número de *bits*, representa o número de

posições diferentes entre as *strings*, por exemplo, o número de posições onde uma *string* possui valores iguais a zero e na outra *string* possui valor igual a um (HAMMING, 1950).

Para a simplificação do circuito digital utilizando MK é necessário agrupar os quadrados que possuem valores iguais a 1, em grupos de potência de 2 $(1,2,4,8,16,\ldots)$, e que sejam vizinhos. Tal qual apresentado na Figura 5(a), são consideradas vizinhas as posições superiores, inferior, à esquerda e à direita do quadrado em questão. O mapa é circular, logo a borda da esquerda faz ligação com a borda da direita e a borda superior faz ligação com a borda superior. Isto permite que um quadrado da borda possa fazer agrupamentos com quadrados de bordas de outras extremidades. Outra possibilidade é que quadrados que estão contidos em um grupo também podem estar contidos em outros grupos,assim como apresentado na Figura 5(b).

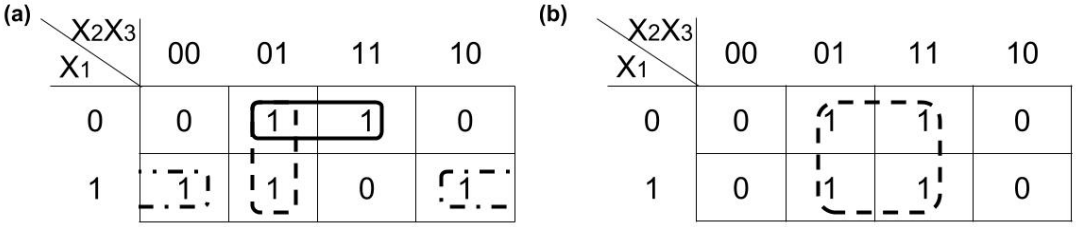



**Figura 5: Exemplos de vizinhanças (a) vizinho na vertical, horizontal e circular (b) grupo de vizinhanças com quatro elementos.**

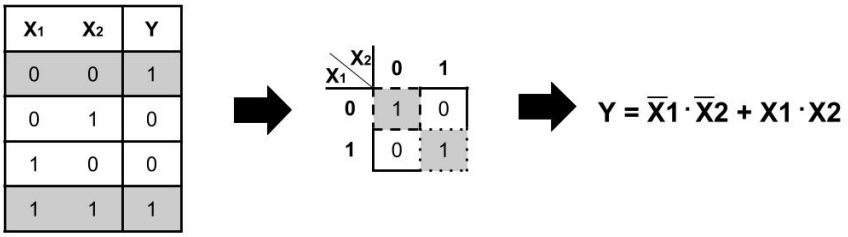
**Fonte: Autoria própria.**

Após agrupar os quadrados o MK pode ser transformado em uma expressão Booleana. As variáveis de cada grupo são ligadas a partir de portas *AND*. No Figura 6(a) é possível observar que existem dois grupos $\overline{x_1}\ \overline{x_2}$ e $x_1\ x_2$, então os grupos são respectivamente $\overline{x_1} \cdot \overline{x_2}$ e $x_1 \cdot x_2$. Para unir os grupos são utilizadas portas *OR* formando a expressão $y = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2$. As variáveis $x$, que mudam de valor não são importantes para a expressão e devem ser omitidas. Como no Figura 6(b), onde no grupo representado pelos quadrados $\overline{x_1}\ \overline{x_2 x_3}$ e $\overline{x_1}\ \overline{x_2} x_3$ o valor de $x_3$ muda, logo ele não será considerado nesta parte da expressão.

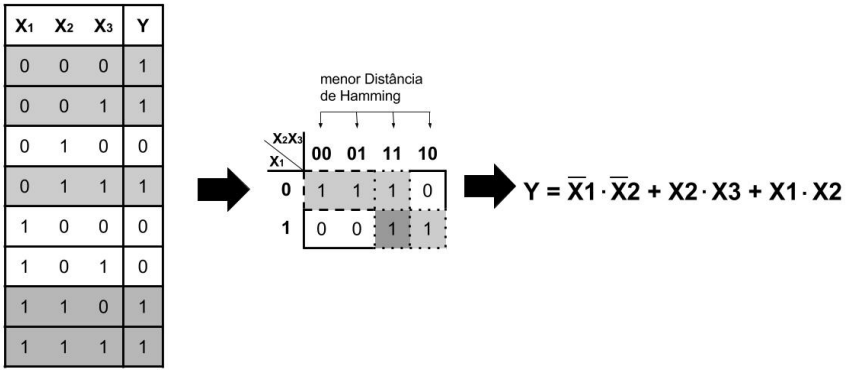
### 2.3.3.2 ALGORITMO DE QUINE-MCCLUSKEY

O Algoritmo de Quine-McCluskey, tal qual o MK, é um método que permite encontrar a expressão Boleana mínima a partir de uma tabela-verdade (QUINE, 1955, 1952).

A primeira etapa para obter a expressão Booleana com o método de Quine-McCluskey é agrupar os estados do conjunto $X$, assim como apresentado na Figura 5(b). Os agrupamentos devem considerar a quantidade de estados em 1, ou seja, o primeiro grupo seria composto por estados do conjunto $X$ que não contenha nenhum 1, como no caso de 000 para três variáveis na




(a) Mapa de Karnaugh com 2 variáveis




(b) Mapa de Karnaugh com 3 variáveis

**Figura 6: Exemplos de Mapa de Karnaugh com 2 e 3 variáveis.**
**Fonte: Adaptado de (GONÇALVES, 2008)**

tabela-verdade. Um segundo grupo um estado 1 do conjunto $X$, como 001, 010 e 100 para o mesmo exemplo, e assim sucessivamente até ter um grupo que todos os valores das variáveis do conjunto $X$ sejam iguais a 1. Outro fator que deve ser respeitado no agrupamento é que a saída do conjunto $X$ deve ser igual a 1 ou $\xi$ (*don't care*).

Após o primeiro agrupamento são feitos novos agrupamentos até que sobre apenas um grupo. O processo de agrupamento é em par e sequencial, ou seja, todos os elementos do grupo 1 tentam se combinar utilizando uma determinada regra com os elementos do grupo 2 e os elementos do grupo 2 com os elementos do grupo 3. A regra para agrupar os conjuntos $X$ é que pelo menos dois estados devem ser iguais e o estado que não é igual recebe o valor $\_$ (*underline*). A representação contínua dos estados discretos (0, 1 e $\_$) é mostrada na Figura 8, onde 1 representa o valor inteiro, *underline* representa o valor contínuo e 0 é neutro não adicionando nenhum valor a representação contínua. No resultado do último grupo é possível observar que dois estados estão repetidos, então um dos estados é desconsiderado. Então, os valores que não se combinaram (não receberam $*$) e que estão no último grupo são os chamados valores primos. Os valores que estão entre parênteses são a apresentação em valor contínuo do seu respectivo conjunto.

Posteriormente, ao processo de identificação dos conjuntos primos é montado um qua-

dro de primos implicantes, que serão considerados para a formação da expressão Booleana. O quadro contém os valores dos primos encontrados pela linha dos valores de saída da tabela-verdade iguais a 1. Na sequência são identificadas quais são as linhas que obedecem aos valores primos encontrados considerando os estados dos conjuntos $X$ da tabela-verdade. Como mostrado na Figura 7(c) apenas os primos $4,1$ e $1,6$ atendem a este critério, então serão os únicos a serem considerados na expressão Booleana.

Obtendo os primos implicantes de interesse, as variáveis $X$ são convertidas para a expressão Boleana respeitando os símbolos 1, 0 e _ , tal qual apresentado na Figura 7(c). O valor 1 representa a própria variável $x_i$, 0 representa a variável $x_i$ negada e _ desconsidera a respectiva variável $x_i$. A porta que liga as variáveis $x_i$ de um mesmo implicante primo é a porta *AND* e a porta que liga os implicantes primos é a porta *OR*.

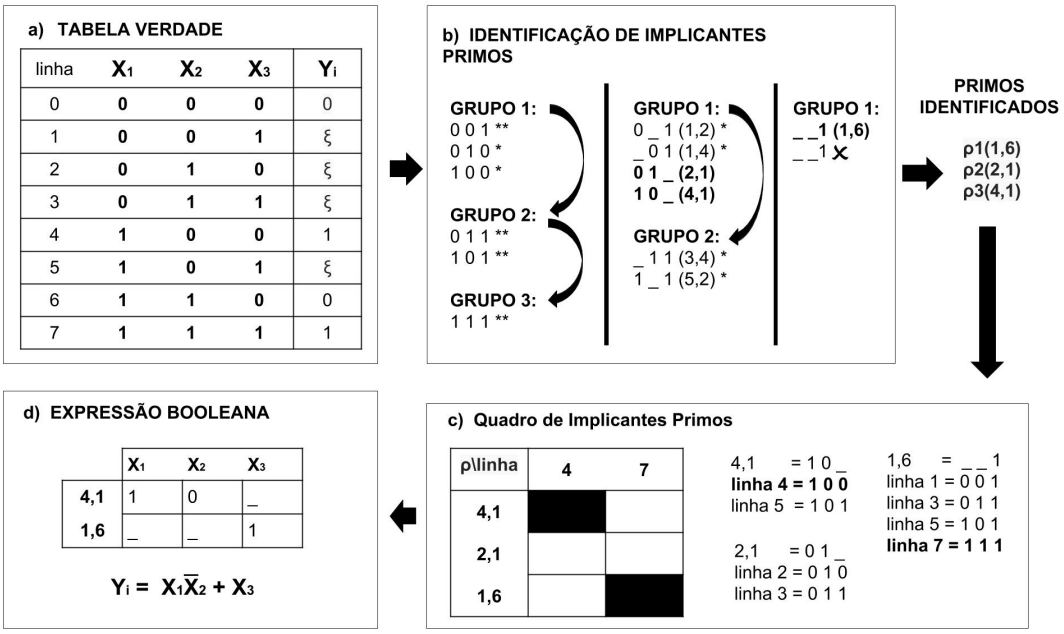



**Figura 7: Método de identificação de expressão Booleana a partir do algoritmo de Quine-McCluskey.** $a$) **apresenta a tabela-verdade na qual será extraída a expressão Boleana.** $b$) **é o processo de agrupamento para identificação dos primos implicantes (**$\rho$**).** $c$) **identifica os implicantes primos que possuem saída** $y_i$ **igual a 1 e que são compatíveis com os** $\rho$ **identificados. Por fim,** $d$) **apresenta a tabela com os primos implicantes que passaram pelo processo anterior e que então definem como as variáveis** $X$ **vão ser ligadas a partir das portas lógicas (*AND*, *OR* e *NOT*).**

**Fonte: Autoria própria.**

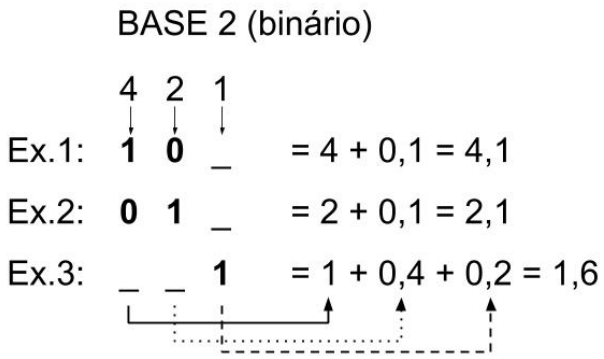



**Figura 8: Representação binária dos exemplos 1, 2 e 3 respectivamente** $10\_$ **,** $01\_$ **e** $\_\_1$**. Para a transformação destes conjuntos para representação contínua os valores iguais a** $1$ **representam os valores inteiros e** $\_$ **representam valores decimais na escala da potência de 2** $(1,2,4,8,\ldots)$**. Os valores** $0$ **não representam nenhum valor. Logo, a representação em valores contínuos é a soma dos valores que são representados por** $1$ **e** $\_$ **.**

**Fonte: Autoria própria.**

## 2.4 REDE DE REGULAÇÃO GÊNICA

Dada que a expressão gênica impacta diretamente e indiretamente no fenótipo de um organismo, o efeito da expressão gênica possibilita uma vasta variação de efeitos no organismo. Desde os fatores mais simples como a cor dos olhos até a produção de proteínas essenciais para diversas vias regulatórias. Alguns destes efeitos no organismo envolvem a expressão de diversos genes. Conforme apresentado na Figura 9, no qual apresenta um sistema de regulação gênica onde o produto biológico da expressão do gene 21 interfere no nível de expressão do gene 9, produzindo a proteína 9 que em conjunto com a proteína 21 implica na ativação do gene 1. O gene 30 quando expresso gera a proteína 30 inibindo a expressão do gene 21. E, a expressão do gene 50 no nível transcriptômico inibe a transcrição do mRNA 30. Considerando a grande quantidade de genes que são expressos simultaneamente e a constante influência tanto de fatores externos quanto internos no processo de expressão, a rede de interações entre os genes torna-se um sistema bastante complexo de identificar.

A representação das influências entre os genes de um organismo é denominado rede de regulação gênica, ou GRN (*Gene Regulatory Network*). Comumente, as GRNs são representadas com um grafo direcional ou como uma matriz de adjacência ($M$), como apresentado na Figura 10. Nesta rede um gene ou um conjunto de genes quando expressos podem causar o aumento, a restrição ou a interrupção da expressão de outro gene. A interação gênica é representada no grafo por uma aresta direcional do gene que influencia para o gene influenciado. Na GRN os genes que interferem no nível de expressão de outro gene são denominados de genes

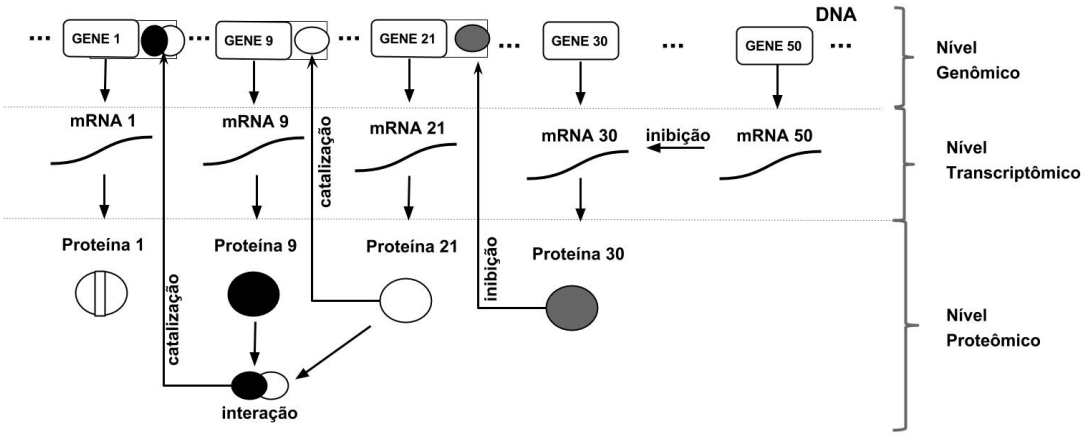



**Figura 9: Representação de uma rede de regulação gênica com 5.**

**Fonte: Adaptado de (HECKER et al., 2009).**

preditores e o gene que sofre a ação do gene preditor é denominado gene alvo.

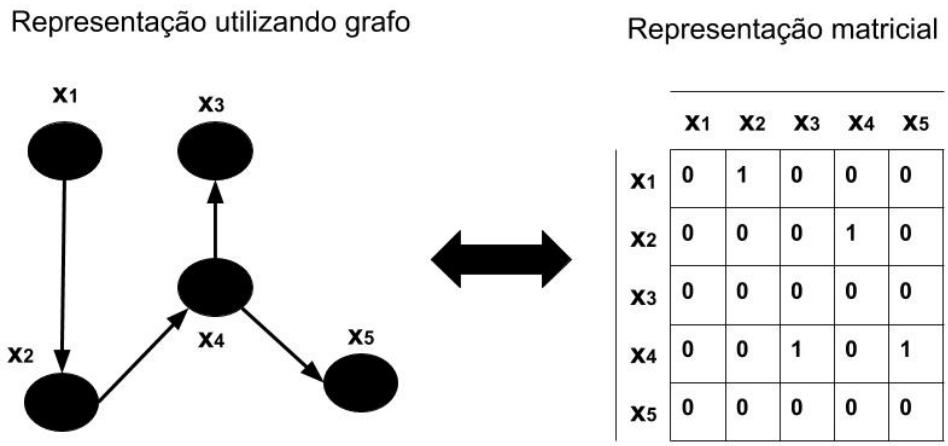



**Figura 10: Representação de uma rede de regulação gênica com 5 genes em forma visual (grafo) e matricial.**

**Fonte: Adaptado de (LOPES, 2011).**

Como as GRNs são uma potencial ferramenta para a análise da interação gênica, elas têm tem sido utilizadas em pesquisas aplicadas a doenças que estão ligadas a algum distúrbio do organismo (HOSSINI et al., 2015). Estes estudos estão diretamente ligados à observação de doenças. Mas também podem ser utilizados para observar o impacto de genes inseridos em um novo genoma, como no caso de estudos em plantas modelos. Outro fator importante é a identificação de genes que possuem um conjunto de genes preditores que indicam um papel relevante no organismo (Martins Junior et al., 2008a; Martins Junior, 2009). Nas Seções 2.4.1 e 2.4.2 são apresentados dois modelos de redes gênicas.

### 2.4.1 REDE BOOLEANA

Kauffman (1969) propôs a aplicação das Redes Booleanas (RB) para a modelagem de GRNs, com o objetivo de simular a dinâmica de sistemas complexos. Estas redes são compostas por um conjunto de vértices $X = \{x_1, x_2, \ldots, x_n\}$, um conjunto de arestas direcionais $E = \{e_1, e_2, \ldots, e_m\}$ e um conjunto de funções Booleanas, também denominadas de funções de Transição Booleana, $\psi = \{\psi_1, \psi_2, \ldots, \psi_n\}$, onde cada função está associada a um $x_i$ (gene i). A função determina o estado binário $[0, 1]$ do gene $x_i$ no próximo tempo, dado os estados binários dos genes preditores no tempo anterior.

A Figura 11 apresenta um exemplo de RB contendo três genes. A Figura 11(a) apresenta a topologia da RB definindo quais são os genes preditores para cada gene alvo, nesta figura $x_1$ é o preditor do gene alvo $x_2$, o gene $x_2$ é preditor do gene $x_3$ e $x_3$ é o gene preditor de $x_1$. Conforme apresentado na Figura 11(b) as funções de transição de estados são compostas por operadores lógicos. Apesar deste exemplo apresentar apenas um operador, as funções podem conter uma combinação de outros operadores. As possíveis variações dos estados de transição é apresentada na Figura 11(c), onde os valores dos *bits* representam sucessivamente os estados dos genes $x_1$,$x_2$ e $x_3$. Na Figura 11(d) é apresentada um exemplo da dinâmica da expressão gênica artificial utilizando as funções de transição Booleanas e um estado inicial arbitrário dos genes da rede.

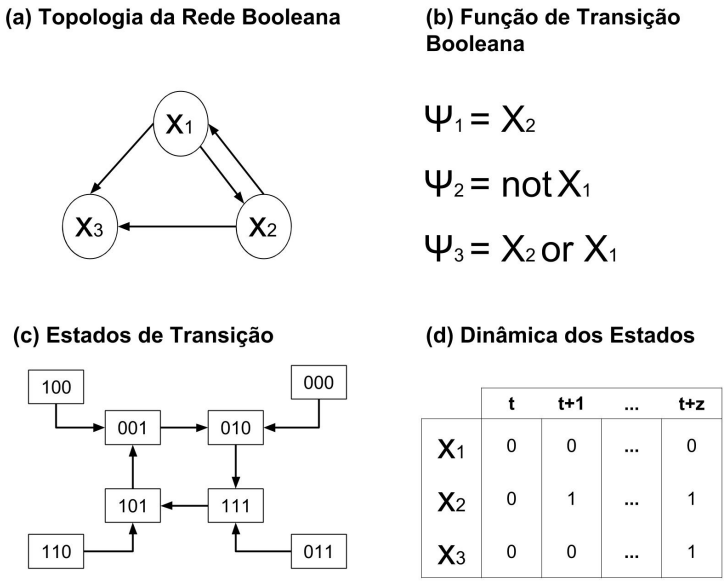



**Figura 11: Exemplo de uma Rede Booleana contendo 3 genes (a) Topologia da rede (b) Funções de transição Booleana (c) Diagrama de transições de estados (d) Dados de expressão gênica artificial.**

**Fonte: Adaptado de (LOPES, 2011).**

No modelo RB as funções de transição Booleana definem os estados dos genes de forma determinística, ou seja, dado o conjunto de estados dos genes da rede a saída será sempre a mesma. Outro fator deste modelo é que as transições dos tempos são discretas. E, a atualização dos estados de todos os genes da rede ocorre de forma síncrona. As funções podem ser compostas por combinações de operadores Booleanos, e a topologia é fixa (não muda no decorrer do tempo). Desta forma a simulação computacional é simplificada e também preserva a dinâmica da rede (KAUFFMAN, 1993a).

Dado ao caráter determinístico da RB os estados de transição é finito, e uma sequencia de estados serão revisitados de forma cíclica. Estes estados são denominados de atratores e as transições que levam até os atratores são denominados estados transientes. A composição dos estados transientes e os atratores formam a bacia de atratores. Os atratores são estados estacionários de um sistema dinâmico. Conforme apresentado na Figura 11(c), os estados 100, 000, 011, 110, são os estados transientes e representam a bacia de atração, enquanto os estados 011 010 111 101 são os atratores.

### 2.4.2 REDE BOOLEANA PROBABILÍSTICA

Um organismo está sujeito a fatores externos que influenciam no sistema de expressão gênica. Neste sentido, além da influencia dos produtos gerados a partir da expressão gênica fatores externos podem interferir na dinâmica do organismo (SHMULEVICH; DOUGHERTY, 2007).

Neste contexto, as RB não representam os fatores externos na dinâmica da expressão. Em contra partida, as RBP permitem inserir os estímulos externos na dinâmica da rede (SHMULEVICH; DOUGHERTY, 2007). Na RBP para cada gene alvo ($x_i$) existem mais de uma função de transição Booleana que podem ser escolhidas a cada instante de tempo. Para cada função de transição Booleana existem uma probabilidade de escolha, sendo que o somatório das probabilidades das funções é igual a um. Apesar de existirem mais de uma função de transição, as entradas dos estados dos genes preditores são os mesmos, ou seja, a topologia é fixa assim como na RB. Uma RBP pode ser interpretada como um conjunto de RBs, onde para cada instante de tempo é selecionada a RB que representa um determinado estímulo do sistema.

Na Figura 12 é apresentado um exemplo da síntese do dado de expressão gênica a partir de um modelo RBP com 3 genes. Neste exemplo é possível observar que existem duas funções de transição para o gene $x_1$, as funções $\vec{\psi}_{1,1}$ e $\vec{\psi}_{1,2}$. A função $\psi_{1,1}$ é composta pelo operador *AND* com probabilidade de 95% de ser escolhida. Enquanto, a função $\vec{\psi}_{1,2}$ é formada pelo operador *OR* com 5% de probabilidade. Observando a dinâmica, a função escolhida no

tempo $t$ é $\vec{\psi}_{1,2}$, contudo é possível observar que no tempo $t+1$ a função $\vec{\psi}_{1,1}$ foi selecionada.

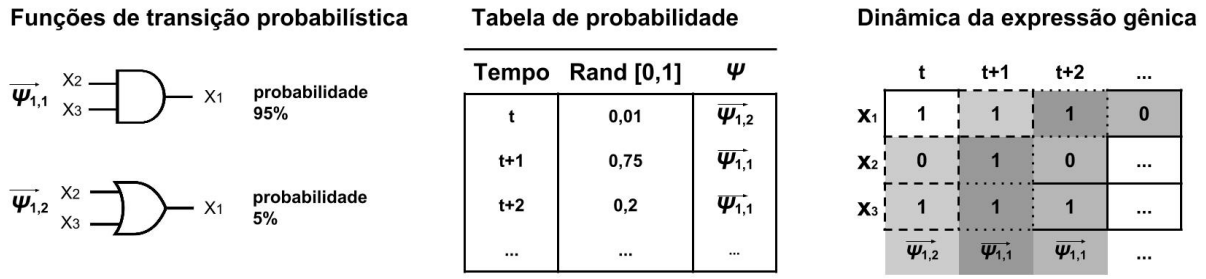



**Figura 12: Exemplo de uma Rede Booleana Probabilística (a) funções Booleanas probabilísticas (b) tabela de probabilidades associada a escolha de uma das funções (c) representação da dinâmica conforme apresentada na tabela de probabilidade.**

**Fonte: Adaptado de (LOPES, 2011).**

## 2.5 COMPUTAÇÃO BIOINSPIRADA

A Computação Bioinspirada é composta por duas grandes áreas que são Computação Evolucionária (CE), que utiliza estratégias baseadas nos princípios da teoria da evolução, e a Inteligência de Enxames (IE), que utiliza estratégias baseadas nos comportamentos sociais de grupos de animais.

Os primeiros trabalhos de CE foram apresentados por Box (1957), Friedberg (1958) e Bremermann (1962) por volta dos anos 50. Entretanto, o início da popularização destes algoritmos teve início a partir do trabalho de (HOLLAND, 1992) apresentando o Algoritmo Genético (AG), Programação Evolucionária (PE) por Fogel (FOGEL, 1962) e Estratégia Evolucionária (EE) por Rechenberg (RECHENBERG, 1965). Estes métodos posteriormente produziram uma gama de aplicações e refinamentos e foram base para outros algoritmos. Os algoritmos de Programação Genética (PG) (KOZA, 1990) e Evolução Diferencial (ED) por Storn e Price (1995) são algumas das novas meta-heurísticas que foram desenvolvidos e que têm despertado o interesse pela comunidade científica.

Os principais métodos da área de IE são: Otimização de Colônia de Formigas (OCF) (DORIGO et al., 2004), Otimização de Enxame de Partículas (OEP) (KENNEDY; EBERHART, 1995), Colônia Artificial de Abelhas (CAA) (KARABOGA; BASTURK, 2008), Algoritmo do Morcego (AM) (YANG, 2010b).

### 2.5.1 EVOLUÇÃO DIFERENCIAL

O algoritmo ED é uma meta-heurística para otimização numérica baseada em operações com vetores. Os candidatos à solução neste algoritmo são codificados por vetores $\vec{\chi}$ contendo valores reais com $NV$ variáveis. Tal qual o AG, o DE contém uma população contendo $NP$ vetores que evolui durante um determinado número $G$ de gerações. O processo de evolução da população ocorre a partir da aplicação do processo de seleção probabilísticas e da ação dos operadores (cruzamento e mutação) sobre os vetores da população. A diferença entre o ED e AG é dada pelo uso de estratégias diferentes de buscas utilizando outras estratégias dos operadores genéticos.

Antes do processo de evolução a população deve ser inicializada de forma aleatória ou respeitando algum critério pré-estabelecido. A população deve ser avaliada através de uma função critério, que gera os valores de qualidade ($fitness$) de cada vetor. A etapa de seleção do DE imita o processo de seleção natural das espécies, onde indivíduos mais adaptados ao meio (melhor *fitness*) possuem maior probabilidade de repassar seu material genético para as próximas gerações. Uma das estratégias de seleção do ED é a seleção por torneio, no qual são escolhidos $\kappa$ vetores da população aleatoriamente e apenas o indivíduo com melhor $fitness$ é selecionado.

Para gerar o vetor testes é necessário criar *a posteriori* o vetor doado que consiste em 3 vetores, denominados vetores pais, $\vec{\chi}_{r1}^{G}$, $\vec{\chi}_{r2}^{G}$, e $\vec{\chi}_{r3}^{G}$, onde $r_1$, $r_2$ e $r_3$ são os índices de vetores distintos da população. Após, a seleção dos vetores pais é aplicado o operador de mutação, como mostrado na Equação 5.

$$\vec{V}^{G} = \vec{\chi}_{r1}^{G} + F(\,\vec{\chi}_{r2}^{G} - \,\vec{\chi}_{r3}^{G}) \tag{5}$$

Onde $\vec{V}^{G}$ representa o vetor doador e $F$ é um valor escalar comumente pertencente ao intervalo [0,4 ... 1]. Na sequência o operador de *crossover* é aplicado ao $\vec{V}^{G}$ e o vetor alvo $\vec{\chi}^{G}$. Este operador mescla as variáveis dos dois vetores gerando uma nova solução, como mostrado na Equação 6.

$$\vec{U}_i^{G} = \begin{cases} V_i^{G}, & \text{se } rand_i <= CR \text{ ou } i = i_{rand}, \\ X_i^{G}, & \text{caso contrário} \end{cases} \tag{6}$$

onde $\vec{U}_i^{G}$ representa o vetor teste após a operação de *crossover* entre o vetor doador e alvo, $CR$ é uma constante que pertence ao intervalo $[0 \ldots 1]$, que define o limiar se $\vec{U}_i^{G}$ recebe o valor de $V_i^{G}$ ou de $\chi_i^{G}$. Por fim, o vetor $U$ é adicionado à população apenas se o valor de seu $fitness$

for menor do que o valor de $fitness$ de $\chi$. O processo apresentado nesta seção é descrito no Algoritmo 1 do Anexo A.

### 2.5.2 COLÔNIA ARTIFICIAL DE ABELHAS

O algoritmo Colônia Artificial de Abelhas (CAA) é inspirado na estratégia de busca por alimentos das abelhas. Esta meta-heurística tem ganhado bastante espaço nas mais diversas áreas (KARABOGA; BASTURK, 2008) inclusive na área de bioinformática (RUBIO-LARGO et al., 2016; GARRO et al., 2016). A ideia deste algoritmo é encontrar fontes de alimento em um espaço de busca, com grandes quantidades de alimento (maiores valores da função *fitness*). A estratégia de busca do CAA considera 3 tipos de abelhas: as abelhas trabalhadoras que exploram o espaço de busca local, as abelhas oportunistas que se baseiam probabilisticamente no *fitness* das abelhas da população para se deslocar no espaço de busca, e as abelhas exploradoras que voam aleatoriamente no espaço de busca sem qualquer tipo de influência. Quando uma abelha encontra um *fitness* melhor do que o já conhecido o exame é influenciado para este local, a partir do processo de seleção por roleta. No caso de uma estagnação do *fitness* durante um período (dado por um limiar de tempo) abelhas trabalhadoras se transformam em abelhas exploradoras para evitar que o algoritmo realize apenas a busca local. O pseudo-código do ABC é apresentado no Algoritmo 2 no Anexo A.

### 2.5.3 ALGORITMO DO MORCEGO

O AM é baseado na eco-localização dos morcegos durante a movimentação em um espaço de busca. Este algoritmo tem sido alvo de diversas pesquisas nas mais diferentes áreas (CORDEIRO, 2013; KRAUSE et al., 2013a; PARPINELLI et al., 2014; MANDAL; KHAN, 2016). A estratégia de eco-localização é baseada no tempo que as ondas ultrassônicas levam do morcego até a fonte (objeto, presa, entre outros), mais o tempo de retorno até o morcego. Quando o objeto está mais próximo do morcego a taxa de pulso ( $r$) aumenta e a amplitude ( $A$) diminui, evitando a perda da localização do objeto. Entretanto, quando o morcego alcança o objeto em questão a amplitude volta a aumentar. Assim, como em todas as meta-heurísticas de IE todos os morcegos sofrem influência da posição do melhor morcego da população (melhor *fitness*). O AM é apresentado no Algoritmo 3 no Anexo A.

## 2.6 MÉTODOS DE BUSCA SEQUENCIAL

Os algoritmos de busca sequencial, assim como os algoritmos bioinspirados, não garantem o ótimo global, entretanto compensam computacionalmente quando comparados ao método exaustivo, principalmente para problemas bastante complexos. Os métodos de buscas sequenciais podem ser classificados como métodos *wrapper*. Basicamente, neste método o processo consiste em adicionar características a um subconjunto, avaliar estas características a partir de um conjunto de dados e uma função critério (função *fitness*) e manter a nova característica apenas se agregar alguma relevância para o subconjunto. No caso onde não exista nenhuma melhora, não reste mais nenhuma característica a ser incluída ou atenda a algum critério de parada o algoritmo é finalizado.

Dada a grande complexidade de encontrar subconjuntos de características em um conjunto muito grande, métodos mais inteligentes foram desenvolvidos como o algoritmo de Busca Sequencial para Frente (BSF) (WHITNEY, 1971), Busca Sequencial para Trás (BST) (MARILL; GREEN, 1963), Busca Sequencial Flutuante para Frente (BSFF) (PUDIL et al., 1994).

### 2.6.1 BUSCA SEQUENCIAL PARA FRENTE

O algoritmo de Busca Sequencial para Frente (BSF) é um método de busca determinística de solução única, o qual é classificado com um algoritmo *wrapper* (GUYON; ELISSEEFF, 2003). Nesta abordagem de busca o subconjunto de características é iniciado sem nenhuma característica $\langle k \rangle = 0$. Quando encontrada a melhor característica avaliada pela função critério, esta é incluída no subconjunto. A seguir, caso seja encontrada uma segunda característica que em conjunto com a primeira produza um resultado ainda melhor, esta segunda característica será adicionada ao conjunto. O algoritmo segue este mesmo processo até atender a um critério. Os critérios de parada podem ser definidos pelo número de características no subconjunto, encontrar um resultado satisfatório, ter percorrido todas as instâncias, ou ter executado um número de iterações.

### 2.6.2 BUSCA SEQUENCIAL FLUTUANTE PARA FRENTE

De forma similar ao BSF, o algoritmo Busca Sequencial Flutuante para Frente (SFFS) é um algoritmo que encontra um conjunto de características sub-ótimo em um espaço de busca (PUDIL et al., 1994). Entretanto, a maior diferença entre o BSF e o BSFF é que este último tem a possibilidade de remover características do subconjunto. Esta habilidade foi introduzida para diminuir o efeito de *nesting*, o qual ocorre no método BSF. Este efeito ocorre quando alguma

característica é adicionada no subconjunto, em razão de melhorar a qualidade do subcojunto no momento. Entretanto, quando tal característica não pertence ao subconjunto ótimo inviabiliza o método a obter o subconjunto ótimo, devido a restrição de quando uma característica é adiciona ao subconjunto numca mais será removida. Após, iniciar com um subconjunto vazio $\langle k \rangle = 0$, o algoritmo de BSFF adiciona uma característica no subconjunto de acordo com a avaliação da função critério, sempre considerando a avaliação do conjunto, assim como no algoritmo de BSF. Quando um número de características predefinidos é alcançado (p.ex. $k = 3$) um processo de remoção é iniciado, ou seja, todas as combinações de $k+1$ menos um dos elementos contidos no subconjunto são testadas. Caso a remoção de uma das características do subconjunto ocasionar uma melhora da avaliação, tal característica é removida do subconjunto. Este processo é aplicado de maneira flutuante até que seja atendido algum critério de parada. O fluxograma do BSFF é apresentado na Figura 13.

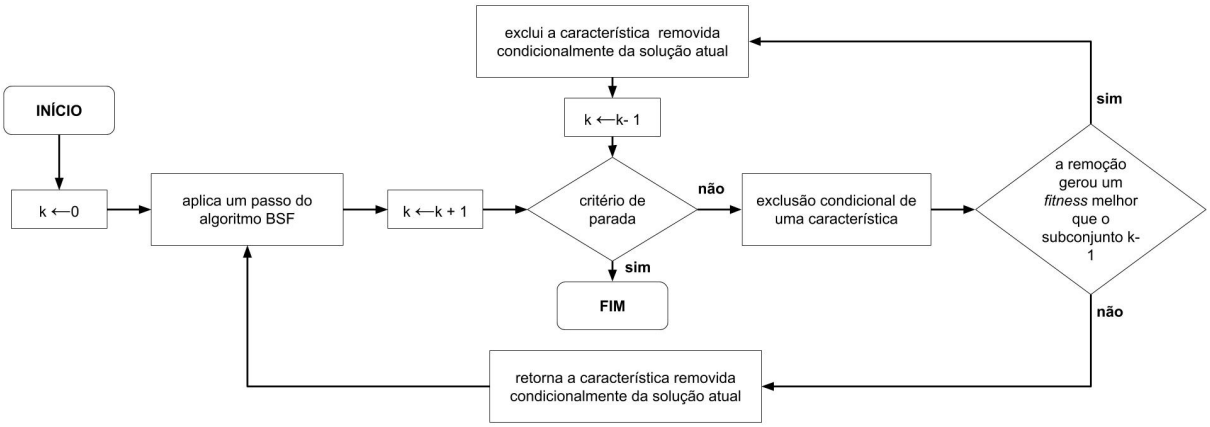



**Figura 13: Fluxograma do algoritmo BSFF. A variável $k$ representa o tamanho do subconjunto que representa a solução atual.**

**Fonte: Adaptado de (SOMOL et al., 2004)**

## 2.7 TEORIA DA INFORMAÇÃO

A área da Teoria da Informação ou Teoria Matemática da Comunicação envolve conceitos matemáticos para representar o termo informação. A área da Teoria da Informação foi unificou com o trabalho publicado por Claude E. Shannon intitulado "*A Mathematical Theory of Communication*"(SHANNON, 1963). Após a popularização da Teoria da Informação sua aplicabilidade foi ampliada sendo atualmente utilizada nas áreas de Estatística, Física, Ciência da Computação, Bioinformática (LOPES et al., 2008; VICENTE et al., 2015; MARBACH et al., 2012), entre outras.

### 2.7.1 ENTROPIA

Normalmente, a entropia é utilizada para quantificar a desordem de um sistema, e também para determinar a quantidade de informação gerada a partir de uma fonte. O conceito inicial foi proposto por Rudolf Clausius na área da termodinâmica (CLAUSIUS, 1879). Posteriormente, Ludwig Boltzmann aplicou a entropia utilizando termos de probabilidades aplicados à área microscópica de um sistema. A entropia desenvolvida neste estudo ficou denominada como, entropia de Boltzmann-Gibbs (BG). Posteriormente Claude Shannon aplicou a entropia na área da Teoria da Informação (SHANNON, 1963).

Da mesma forma que a entropia BG, a entropia de Shannon é definida em termos de probabilidade, como mostrados nas Equações 7 e 8 (LOPES, 2011).

$$H(X) = -\sum_{x \in X} P(x) \cdot \log_2 P(x), \tag{7}$$

onde

$$\sum_{x \in X} P(x) = 1. \tag{8}$$

A Equação 7 apresenta que a quantidade de informação de uma fonte é dada a partir da soma de todas as probabilidades da ocorrência do evento $x$, multiplicado pelo seu próprio logaritmo.

Neste contexto a entropia apresenta a medida de incerteza da variável. Quanto maior este valor, maior é o valor da incerteza desta variável. Da mesma forma que pode ser analisada a incerteza de uma variável, também é possível analisar o nível de incerteza entre duas variáveis de forma condicional, definida como entropia condicional, de acordo com a Equação 9 (LOPES et al., 2011).

$$H(Y \mid x) = -\sum_{y \in Y} P(Y|x) \log_2 (Y|x), \tag{9}$$

tal que $P(Y|x)$ representa a distribuição de probabilidades da variável $Y$ dada a variável $x$. Quando a entropia condicional apresenta um baixo valor, a variável $x$ possui uma grande probabilidade de predizer a variável $Y$.

Conforme apresentado por (Martins Junior et al., 2008b) a Entropia Condicional Média (ECM) é uma média ponderada das entropias condicionais de $Y$ dado a probabilidade de cada um de $x$. , como mostrado na Equação 10.

$$H(Y \mid X) = \sum_{x \in X} P(x) H(Y|x), \tag{10}$$

no qual $H(Y \mid x)$ representa a entropia condicional da variável $Y$ dado $x$ (veja Equação 9). Na função $H(Y \mid X)$ quanto menor valor da entropia de $Y$ dado o conjunto $X$ maior será o ganho da

informação, e consequentemente maior probabilidade do conjunto $X$ predizer $Y$. A Informação Múltua (IM) é a diferença entre a entropia *a priori* e a entropia condicional média, assim como apresentado na Equação 10.

$$IM(X,Y) = H(Y) - H(Y \mid X) \tag{11}$$

## 2.8 TRABALHOS CORRELATOS

Nesta seção são apresentados alguns trabalhos que contribuíram para esta pesquisa. Os trabalhos foram divididos em 4 partes, na Seção 2.8.1 são apresentados trabalhos voltados à síntese de GRNs. Na Seção 2.8.2.1 são apresentados trabalhos aplicados aos métodos de seleção de características e na Seção 2.8.2 são apresentados diferentes métodos de inferência.

### 2.8.1 MÉTODOS DE SÍNTESE DE GRNS

Um *framework* que gera redes artificiais para simular dados de expressão gênica são apresentados em Lopes et al. (2008, 2011). Neste *framework* foram consideradas as topologias de Erdös-Rényi, Watts-Strogatz e Barabási-Albert. Este trabalho mostra que com o aumento da média de ligações, a precisão diminui e também mostra que a quantidade de dados de expressão gênica representa um fator importante para os métodos de inferência.

Schaffter et al. (2011) também propuseram um software que permite a geração de dados de expressão gênica para avaliação de métodos de inferência, denominado GeneNetWeaver [1]. Este algoritmo permite gerar dados *in silico* de organismos como E. *coli* e S. *cerevisiae*. Este *framework* foi utilizado diversas vezes no projeto *Dialogue on Reverse Engineering Assessment and Methods*[2] (DREAM).

Em Terfve et al. (2012) é apresentado um modelo *open-source* implementado em linguagem R para a simulação de dados de expressão gênica, denominado CellNOptR[3]. Este modelo permite a utilização de modelos baseados em rede Booleana, *Fuzzy*, e Equações Diferenciais para gerar os dados de expressão.

Em (CHAI et al., 2014) é apresentado uma revisão sobre métodos para construir GRNs, dividindo-os em seis classes: redes Booleanas, redes Booleanas probabilísticas, equações diferenciais, redes neurais, redes Bayesianas e redes Bayesianas dinâmicas. Também são apresen-

[1] fonte acessada em 25 de Fevereiro de 2016: http://gnw.sourceforge.net/
[2] fonte acessada em 25 de Fevereiro de 2016: http://dreamchallenges.org/
[3] fonte acessada em 25 de Fevereiro de 2016: http://www.cellnopt.org/

tados vantagens e desvantagens de cada método.

### 2.8.2 MÉTODOS DE INFERÊNCIA DE GRNS

Liang et al. (1998) apresenta o algoritmo *REVerse Engineering ALgorithm* (REVEAL) para a inferência de GRNs. Esta abordagem infere os relacionamentos entre os genes utilizando a informação mutua (veja Equação 10), considerando um gene alvo Y e um conjunto de genes preditores X. Neste método o objetivo é buscar um subconjunto X que determina completamente o comportamento de Y, o tamanho do subconjunto de X é um parâmetro de entrada desta abordagem. Segundo Kelemen et al. (2008) como $H(X) = H(Y,X)$, logo $I(X,Y) = H(Y)$, desta forma não é necessário calcular explicitamente a informação mutua $I(X,Y)$ tornando o calculo mais rápido. (AKUTSU et al., 2003) apresenta uma prova matemática que com um pequeno número estados de transição (dados de expressão gênica) é possível recuperar uma RB corretamente, conforme apresentado por Liang et al. (1998).

No artigo de Sakamoto e Iba (2001) é apresentado um resumo de métodos para a inferência de GRNs e métodos menos tradicionais que utilizam analises de interações entre proteínas com outras moléculas e genes. São apresentados análises de paradigmas de engenharia reversa incluindo modelos discretos, contínuos e híbridos que podem ser utilizados para a recuperação da estrutura das redes.

Iba e Mimura (2002) apresentam uma abordagem de inferência de rede de regulação gênica utilizando dados de expressão gênica utilizando AG utilizando dados temporais de expressão gênica. Os resultados indicaram que a abordagem bioinspirada apresentou estruturas precisas utilizando poucas amostras.

Foi apresentado por Margolin et al. (2006) o *Algorithm for the Reconstruction of Accurate Cellular NEtworks* (ARACNE) que se baseia em redes de alta relevância. Neste método a medida de dissimilaridade e similaridade é comumente definida por meio de correlação ou informação mútua. Posteriormente ao processo de limiarização, é aplicado um tratamento entre os vértices da rede inferida, denominado *data processing inequality*. Neste tratamento, são selecionados triplas de genes $(x_1, x_2, x_3)$ que são dependentes, onde caso a similaridade de $x_1 \rightarrow x_2$ for menor do que $x_2 \rightarrow x_3$ e $x_1 \rightarrow x_3$ a conexão de $x_1 \rightarrow x_3$ é eliminada.

O algoritmo *Context Likehood of Relatedness* (CLR) apresentado em Faith et al. (2007) é uma extensão da abordagem das redes de alta relevância assim como o algoritmo ARACNE. Este método adota a informação mútua como medida de similaridade entre dois genes e define se os pares de genes estão conectados através de um limiar.

Meyer et al. (2007) apresenta um método de seleção de características por máxima relevância / mínima relevância denominado de MRNET. Nesta abordagem para cada gene alvo $Y$ é escolhido o gene preditor $x$ que possui o maior valor de informação mutua em relação ao gene alvo $I(x_i,Y)$. O segundo gene preditor $x_j$ é o gene que possui a maior informação mutua em relação ao gene alvo $I(x_j,Y)$ e que possua a menor informação mutua em relação ao gene escolhido anteriormente $I(x_i,x_j)$ gerando o subconjunto $Z$. Os próximos genes preditores adicionados terão que maximizar a diferença $u_i - r_i$, no qual $u_i$ é dado por $I(x_i,Y)$ e $u_i$ por $r_i = \frac{1}{|Z|}\sum_{x_j \in Z} I(x_i,x_j)$.

(XU et al., 2007) apresenta um método de inferência de GRNs utilizando uma Rede Neural Recorrente (RNR) treinada a partir de algoritmos bioinspirados, os algoritmos ED, OEP e um algoritmo híbrido utilizando as duas abordagens Evolução Diferencial Otimização por Enxame de Partículas (EDOEP).

Um método baseado no algoritmo de floresta aleatória para inferência de redes de regulação gênica é apresentado em Huynh-Thu et al. (2010). Este algoritmo denominado GENIE3[4] divide o problema de inferência de GRNs em diversos subproblemas, assim como apresentado em Jimenez et al. (2015). Porém este método não é de seleção de característica e sim de regressão. Este método apresentou bons indicativos com dados sintéticos e biológicos, mostrando ser um método bastante eficiente e genérico.

Marbach et al. (2012) apresentam diversos métodos para a inferência de GRNs através do projeto DREAM. Neste trabalho foram analisados o desempenho de 35 métodos de inferência individualmente em 3 redes biológicas (S. *cerevisiae*, E. *coli* e S. *aureus*). Também foi utilizada uma abordagem populacional que agrega as informações de todos os métodos e monta uma rede a partir de votações entre os 35 métodos propostos. Esta abordagem apresentou melhorias em quase todos as redes biológicas em relação aos 35 métodos separadamente.

Gallo et al. (2013) apresentam uma revisão das abordagens de Regras de Associação (RA) para extrair informações relevantes dos genes. Este trabalho apresenta o estado da arte dos principais tópicos do método de RA aplicados a engenharia reversa de GRNs e das principais técnicas recorrentes.

Jimenez et al. (2015) apresentam uma abordagem utilizando o AG aplicado à inferência de GRNs. Nesta abordagem o AG não é executado para toda a rede e sim para cada gene da rede. Esta abordagem utilizou o Critério de Informação de Akaike (CIA) como função *fitness*. Os resultados apresentados indicaram a adequação do critério adotado, mesmo quando o número de amostras é baixo.

[4]fonte acessada em 16 de Abril de 2016: http://migre.me/tygH5

(HATTORI et al., 2015) apresenta uma comparação entre um algoritmo de CE, o algoritmo EDD, com duas abordagens de algoritmos de busca sequencial, o BSF e BSFF. Nesta abordagem foram utilizados RBP para gerar a dinâmica da rede com topologias ER, WS e BA. Os resultados aprestados mostraram que em média o algoritmo EDD obteve melhores medidas de precisão em relação aos métodos de busca sequenciais.

#### 2.8.2.1 MÉTODOS DE SELEÇÃO DE CARACTERÍSTICAS PARA INFERÊNCIA DE GRNS

No trabalho de Martins Junior (2009) é apresentada uma abordagem de seleção de características utilizando a entropia condicional como função critério (*fitness*). Neste trabalho, foi proposto o uso de redes Booleanas probabilísticas, e também análises biológicas de dados de *microarray* de um agente causador da malária, o *Plasmodium falciparum* (BARRERA et al., 2005, 2006, 2007). Também foram realizadas análises sobre a Predição Intrinsecamente Multivariada (PIM) (MARTINS et al., 2008; Martins Junior et al., 2008a) de conjuntos de 2 e 3 características preditivas relacionando com uma variável alvo. Este estudo mostrou a importância dos genes alvos que possuem estes conjuntos PIM, como no caso do gene DUSP, particularmente ligado ao câncer de pele (melanoma). Além, de demostrar que existem genes alvos que possuem características fundamentais para vias metabólicas.

No trabalho de Lopes et al. (2010) é apresentada uma abordagem de seleção de características para inferência de GRNs. Nesta abordagem é utilizado um conhecimento a priori na estratégia de busca, tal algoritmo é denominado SFFS-MR. Comparado aos algoritmos BSF e BSFF, o algoritmo SFFS-MR apresentou melhores valores de acurácia.

O algoritmo SFFS-BA é apresentado no trabalho de Lopes et al. (2014). Tal processo, é derivado do algoritmo BSFF e que se diferencia incluindo conhecimento de informações estruturas *a priori*. Os resultados para a inferência de redes gênicas artificiais mostraram melhores resultados quando comparados com os algoritmos BSF e BSFF.

Vicente et al. (2015) apresento uma abordagem de seleção de características para inferência de GRNs utilizando integração de dados biológicos aplicados a A. *thaliana*. Este trabalho utiliza o algoritmo determinístico BSFF e função critério ECM. Os resultados deste trabalho mostraram indícios de progresso no método de inferência quando inseridas informações de localização celular.

# 3 MATERIAIS E MÉTODOS

Este trabalho propõe uma abordagem de seleção de características utilizando algoritmos de Computação Bioinspirada e Busca Sequencial para a inferência de Redes de Regulação Gênica utilizando dados temporais de expressão gênica. Além disso, são apresentados os métodos de otimização da rede inferida utilizando o algoritmo de Quine-McCluskey (QM) e rede de consenso.

O processo geral aplicado neste trabalho é apresentado na Figura 14. Neste trabalho os dados temporais de expressão gênica são obtidos a partir de uma Rede Gênica Artificial (AGN), tal qual apresentado na Figura 14(a). Para gerar os dados artificiais de expressão gênicas por sua vez é necessário criar a rede que representa as ligações entre os genes preditores de cada gene alvo e as funções de transição que definem as expressões dos genes alvos.

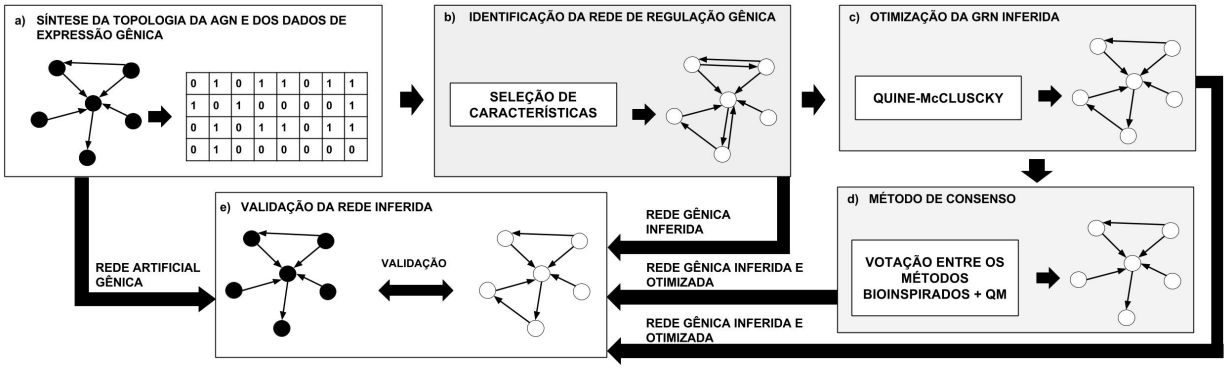



**Figura 14: Processo geral de inferência de redes de regulação gênica. (a) representa a síntese da interação entre os genes da rede, (b) representa o modelo de seleção de características, (c) é a etapa de otimização da rede inferida utilizando o algoritmo de QM, (d) apresenta a otimização da rede utilizando o consenso entre as redes inferidas com a otimização por QM (e) comparação entre a rede inferida e a rede real.**

**Fonte: Autoria própria.**

A inferência de GRNs atualmente é um dos grandes desafios na área de bioinformática, devido ao grande número de características e ao pouco número de amostras. Neste contexto este trabalho propõe uma abordagem de seleção de características utilizando algoritmos bioinspira-

dos e de busca sequencial para a identificação de genes preditores para cada gene da rede, tal qual apresentado na Figura 14(b).

Na Figura 14(c) é proposta a otimização da rede inferida utilizando o método de QM. Este método permite obter a expressão Booleana mínima, removendo variáveis (genes preditores) que não são importantes. Também é apresentado na Figura 14(d) outro pós-processamento utilizando os resultados das inferências de três algoritmos bioinspirados com a otimização por QM para gerar uma rede de consenso.

Na última etapa no processo de inferência, apresentada na Figura 14(e), é a validação da rede inferida. Nesta etapa é comparado a rede real, a qual foi base para gerar os dados de expressão gênica, com a rede inferida.

Esta seção é organizada da seguinte forma: Na Seção 3.1 é apresentado o processo de síntese do modelo de topologia e síntese dos dados de expressão da AGN. Na Seção 3.2 são apresentados os métodos propostos neste trabalho. Na Seção 3.4 é apresentado o método de validação dos desempenhos dos algoritmos.

## 3.1 SÍNTESE DA TOPOLOGIA E DOS DADOS DE EXPRESSÃO GÊNICA

Nesta seção será apresentada a síntese das AGNs e como são gerados os dados de expressão gênica temporal. O modelo apresentado é baseado no software jAGN (do inglês *Java-Based Model for Artificial Gene Networks Generation* (LOPES et al., 2011).

O mínimo para avaliar o método de inferência de GRNs é necessário conhecer *a priori* como é a estrutura da rede do organismo e ter uma quantidade de amostras de dados de expressão gênica equivalentes a quantidade de genes que vão ser analisados. Atualmente, estes cenários são muito difíceis de encontrar. Neste contexto, métodos computacionais que permitem simular os dados de expressão gênica e fornecer a estrutura da rede a partir da qual foram gerados os dados de expressão gênica assumem um papel muito importante na área de inferência.

Na seção 3.1.1 e na Seção 3.1.2 serão apresentadas como foi realizada o processo da síntese da topologia da AGN e dos dados de expressões gênicas artificiais.

### 3.1.1 TOPOLOGIA DA REDE DE REGULAÇÃO GÊNICA

Uma AGN é descrita formalmente como $G = \{X, E, S, \Psi\}$, no qual os conjuntos $X$ e $E$ representam as características de topologia das AGNs, enquanto os conjuntos $S$ e $\Psi$ utilizam

a topologia para gerar os dados de expressão temporal. Neste contexto, o conjunto $X$ é descrito como $X = \{X_1, X_2, \ldots, X_n\}$, onde $x_i$ representa o gene $i$ e $n$ é o número de genes (vértices) da rede. O conjunto $E$ é representado por $E = \{e_1, e_2, \ldots, e_m\}$, onde $m$ representa o número de arestas da rede. Cada elemento do conjunto $E$ é um par ordenado representado por $e = \{i, j\}$, no qual as variáveis $i, j \in \{1, 2, \ldots, N\}$ descrevem a direção da aresta que parte do gene $x_i$ (preditor) e incidência no gene $x_j$ (alvo). Os conjuntos $X$ e $E$ de uma AGN podem ser representados por uma matriz de adjacência $M$ (veja Figura 10), assim como os modelos de redes complexas. Na matriz de adjacência as ligações entre os genes $x_i \rightarrow x_j$, representadas pelo par ordenado $e = \{i, j\}$, recebem $M(i, j) = 1$, e enquanto não há interação, quando não houver interações $M(i, j) = 0$.

Os modelos teóricos de redes complexas foram utilizados para definir as características topológicas das AGNs utilizadas neste trabalho, conforme (LOPES et al., 2011). Dentre os modelos de redes complexas, os modelos adotados para a caracterização das topologias das AGNs foram: *Uniformly-Random* (ERDÖS; RÉNYI, 1959) (ER), *Small-World* (WATTS; STROGATZ, 1998) (WS) e *Scale-Free* (BARABÁSI; ALBERT, 1999) (BA) (veja Seção 2.2). A escolha do modelo topológico (ER, WS e BA), a quantidade de genes ($n$) e a média do grau de entrada de cada gene da rede ($\langle k \rangle$) são os parâmetros necessários para a criação da topologia. Depois de gerada a topologia da rede é formada uma matriz $M$ simétrica, ou seja, a posição $M(x_i, x_j)$ é igual a $M(x_j, x_i)$ contendo $2k$ arestas. Para transformar $M$ em uma matriz direcional é removida uma das posições onde $M(x_i, x_j) = 1$ com probabilidade de 50%. Produzindo, assim, o $\langle k \rangle$ de ligações entre os genes da rede.

### 3.1.2 EXPRESSÃO GÊNICA ARTIFICIAL

Após a síntese da topologia da AGN é gerado o conjunto $S = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_z\}$, que representa os $Z$ estados binários da rede (expressão gênica artificial), cada elemento do conjunto $S$ é um vetor $\vec{s}_t$ que armazena os $n$ estados (nível de expressão) dos genes no instante de tempo $t = \{1, 2, ..., Z\}$, assim como apresentado Figura 15. As transições de estados são definidas a partir do conjunto $\Psi = \{\Psi_1, \Psi_2, \ldots, \Psi_n\}$, no qual cada elemento deste conjunto é descrito por um circuito lógico que representa a transição do estado ($\vec{s}_{t,j}$) do gene $x_j$ para o próximo estado ($\vec{s}_{t+1,j}$).

Este trabalho utiliza o modelo Rede Booleana Probabilística para gerar as transições de estados (veja Seção 2.4). Cada $\Psi_i$ é composto por um conjunto de circuitos lógicos que são escolhidos de forma probabilística, onde um dos circuitos possui uma probabilidade dominante em relação aos outros circuitos, neste trabalho foram adotados os parâmetros padrões da ferra-

| | $S_1$ | $S_2$ | ... | $S_z$ |
|---|---|---|---|---|
| $G_1$ | 1 | 0 | ... | 0 |
| $G_2$ | 0 | 0 | ... | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $G_n$ | 1 | 1 | ... | 0 |

**Figura 15: Exemplo de um conjunto de expressões gênicas artificiais binárias contendo $n$ genes e $z$ tempos de expressão gênica.**

**Fonte: Autoria própria.**

menta jAGN de 96% para o circuito mais dominante e 4% distribuido entre os outros circuitos, e cada gene possui no total 3 funções de transição. Este método é adotado para simular o sistema ruidoso biológico. Para gerar os circuitos deste modelo foram utilizados os seguintes operadores: $NOT, AND, OR, NAND, NOR$ e $XOR$. Sendo que também é possível utilziar a combinação entre as portas lógicas para gerar a função de transião. As entradas dos circuitos são os estados dos genes preditores ( $\vec{s_t}$) do respectivo gene alvo.

A síntese de cada circuito lógico é dada por uma intercalação entre um gene preditor e uma porta lógica, sendo que as portas são escolhidas de forma probabilística. Na Figura 16 é apresentado um exemplo de uma síntese, onde os níveis de expressões dos genes preditores um e dois interagem a partir de uma porta lógica *AND* e a saída deste operador é a entrada de uma porta *OR* com a entrada da expressão do gene 3, resultando na expressão do gene alvo 4.

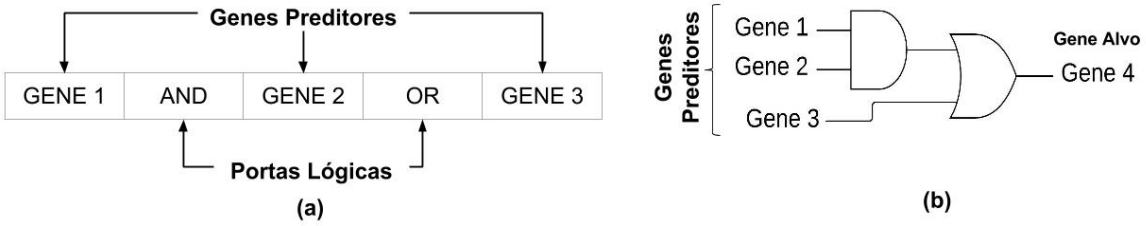



**Figura 16: Exemplo de uma síntese de um circuíto lógico, onde os genes 1, 2 e 3 são genes preditores e 4 é o gene alvo (a) modelo em vetor (b) modelo em diagrama.**

**Fonte: Autoria própira.**

## 3.2 INFERÊNCIA DA REDE DE REGULAÇÃO GÊNICA

Neste método a identificação é feita utilizando a entropia como medida de identificação dos padrões de similaridade entre os genes, e os métodos de buscas aplicados a este trabalho são algoritmos de busca bioinspirados e sequenciais, veja respectivamente as seções 2.5 e 2.6.

A identificação das GRNs se torna uma difícil tarefa considerado a grande dificuldade de encontrar informações biológicas de expressão gênica, a alta complexidade de um sistema biológico, as falhas no processo de identificação de expressão gênica e aos poucos números de amostras de experimentos. Ao invés de apresentar a estrutura detalhada das GRNs, este trabalho tem o objetivo de apresentar um método que apresente ao final do processo de identificação as ligações entre os genes da rede de forma global.

Dentre os modelos de identificação de redes de regulação gênica este trabalho utiliza uma técnica de reconhecimento de padrões, denominada seleção de características. Nesta abordagem, o problema é dividido em $n$ subproblemas de seleção de características, tal qual apresentado na Figura 17. Para cada solução do *i-ésimo* subproblema são obtidas as características mais relevantes, ou os genes preditores mais relevantes, para o *i-ésimo* gene. E, assim que são realizados todos os $n$ subproblemas apenas os conjuntos de genes que passam por um *threshold* são os que devem ser considerados na rede, o valor de *threshold* de $0,3$ foi apresentado por Lopes et al. (2008).

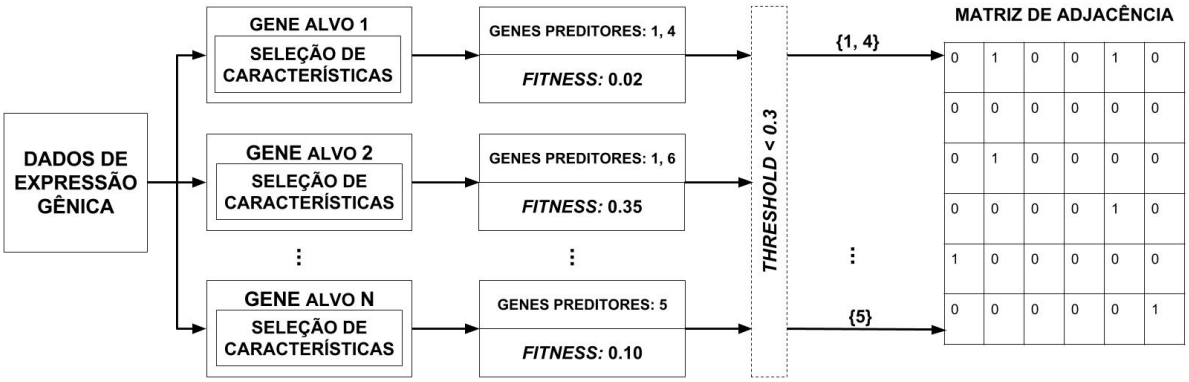



**Figura 17: Processo de inferência de redes de regulação gênica utilizando o método de seleção de características.**

**Fonte: Adaptado de (JIMENEZ et al., 2015).**

No modelo de seleção de características são utilizados um algoritmo de busca e uma função critério. Os algoritmos utilizados neste trabalho serão apresentados sequencialmente.

### 3.2.1 CODIFICAÇÃO E DECODIFICAÇÃO DOS ALGORITMOS BIOINSPIRADOS

Neste trabalho são apresentadas duas codificações para as possíveis soluções na inferência de GRNs utilizando variáveis contínuas. Segundo Krause e Lopes (2013) e Krause et al. (2013b) a abordagem contínua para aplicado a problemas discretos tem apresentado bons indicativos quando comparado com as abordagens binárias, devido a evolução ser mais suave,

evitando as mudanças bruscas que ocorrem na abordagem binária.

As abordagens dos algoritmos bioinspirados utilizados neste trabalho, possuem vetores ou partículas que possuem valores contínuos que são inicializados com valores aleatórios entre os valores de $[-1 \ldots 1]$. Os valores são discretizados a partir da função sigmóide, como mostrado na Equação 12, tal função foi escolhida por ser uma das mais eficientes e exploradas na literatura (BANATI; BAJAJ, 2011; PALIT et al., 2011; KRAUSE et al., 2013b).

$$\vec{\chi}_i = \begin{cases} 1, & \text{se } \frac{2}{1+e^{-2x_i}} - 1 > 0, \\ 0, & \text{caso contrário} \end{cases} \tag{12}$$

onde $\vec{\chi}_i$ é um elemento discretizado do vetor.

A primeira abordagem codifica cada posição do vetor em um gene, tal qual apresentado na Figura 18. O vetor nesta abordagem possui tamanho $NV = n - 1$, sendo $n$ o número de genes e $NV$ o número de variáveis do vetor, desconsiderando o gene alvo na busca. Para obter o subconjunto dos índices dos genes preditores, cada posição do vetor é discretizada. Posteriormente, cada elemento $i$ do vetor corresponde ao *i-ésimo* gene que está fazendo ligação (1) ou não (0) para o gene alvo formando o subconjunto que é avaliado pela função critério.

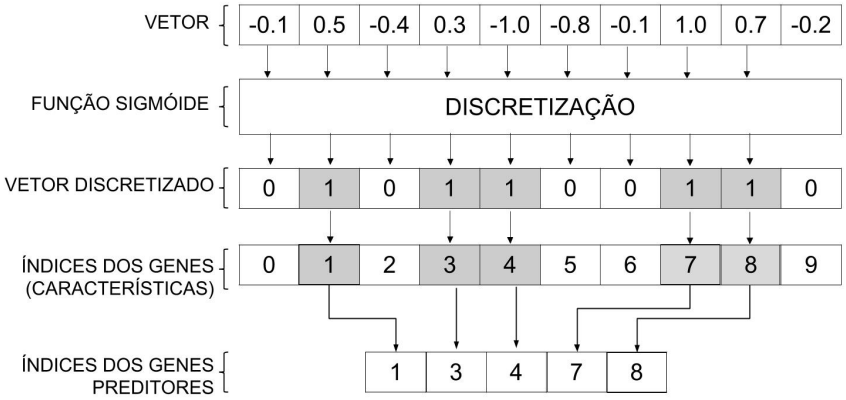



**Figura 18: Processo de decodificação dos índices dos genes preditores (características) do $\chi_i$. Nesta abordagem cada posição do vetor representa o índice de um gene preditor, ou seja, $n-1$ posições, dado que não é considerada a autorregulação do gene alvo.**

A segunda abordagem permite restringir o número de características que pode ser condifcada no vetor, nesta abordagem um conjunto de *bits* do vetor codifica um índice de um gene preditor, tal qual apresentado na Figura 19. Baseado em Kauffman (1993b), que afirma que a fronteira entre a ordem e o caos é de aproximadamente 3 conexões por nodo, então foi definido um tamanho máximo de 4 genes preditores que poderiam ser codificados por vetor. Para cada conjunto de *bits* foi adicionado um *bit* a mais, que indica se seu respectivo índice está ativado ou não. Desta forma é possível um vetor codificar entre 0 e 4 genes preditores. Assim como na primeira abordagem esta abordagem codifica os índices dos preditores em valores contínuos e

que são posteriormente discretizados utilizando a função sigmóide, apresentada na Equação 12. Depois da discretização os conjuntos de *bits* são decodificados para índices que podem estar ativos ou não dependendo do valor do *bit* extra para cada conjunto.

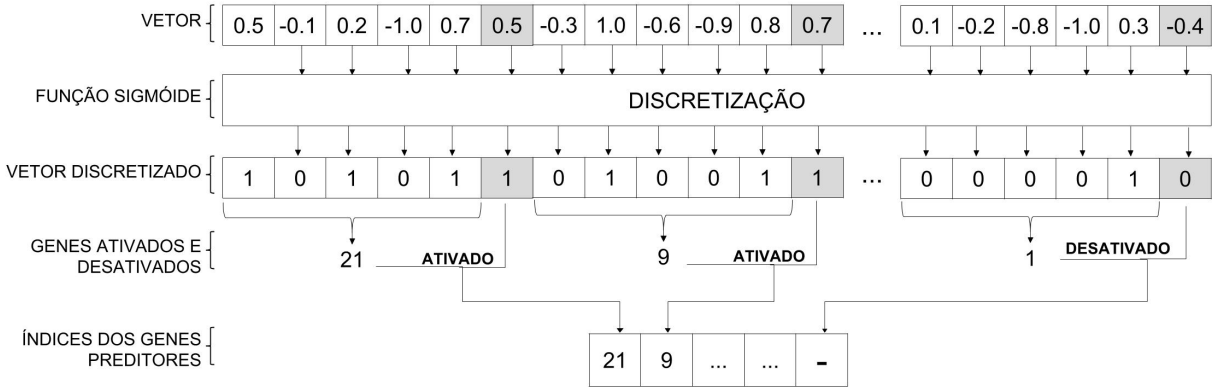



**Figura 19: Processo de decodificação dos índices dos genes preditores (características) do $\chi_i$. Nesta abordagem um conjunto de *bits* representa um índice de um gene preditor. Existe também um *bit* extra para cada conjunto de *bits* que permite ativar ou desativar seu respectivo conjunto.**

### 3.2.2 FUNÇÃO CRITÉRIO

A função critério neste trabalho tem o objetivo de avaliar subconjuntos de genes preditores em relação a um determinado gene alvo, utilizando o conjunto de dados temporais de expressão gênica e apresentar um valor de quanto este subconjunto define o gene alvo. Neste trabalho foi adotada a função critério Entropia Condicional Média (ECM), apresentado na Equação 10 da Seção 2.7.1. Esta função permite identificar estatisticamente genes preditores $X$ em relação a um gene alvo $Y$ (CHARBONNIER et al., 2010). A ECM funciona em conjunto com o algoritmo de busca, que recebe os sub-conjuntos e indica, a partir do valor do ECM, o quão dependetes são as variáveis.

Uma ilustração da tabela de frequências absolutas (número de ocorrências) de $Y = x_3$ dado cada uma das instâncias de $X_1$ $X_2$ e de $X_4$ $X_5$ pode ser observada na Figura 20. Esta tabela é transformada em uma distribuição conjunta de probabilidades, a qual posteriormente será usada para obter a ECM de $Y$ dadas as variáveis $X$. Na Figura 20(a) é possível observar que bons preditores possuem uma baixa entropia. Uma baixa entropia significa que o conjunto $X$ representa bem a variável $Y$, como é possível observar na tabela de frequência, onde as classes estão bem concentradas em uma das classes em todas as variações do conjunto $X$. Na Figura 20(b) é possível observar que as distribuições da tabela de frequência estão mais próximos da distribuição uniforme para todas as instâncias do conjunto $x$, indicando que o conjunto $X$ não define inequivocamente a variável $Y$. Neste caso o valor de entropia é elevado.

| $X_1$ | $X_2$ | $Y = X_3$ | |
|---|---|---|---|
| | | 0 | 1 |
| 0 | 0 | 0 | 9 |
| 0 | 1 | 8 | 0 |
| 1 | 0 | 0 | 7 |
| 1 | 1 | 4 | 1 |

- Baixa Entropia;
- Frequência bem concertada em uma das classes;
- **Bons preditores**

**(a)**

| $X_4$ | $X_5$ | $Y = X_3$ | |
|---|---|---|---|
| | | 0 | 1 |
| 0 | 0 | 4 | 5 |
| 0 | 1 | 3 | 5 |
| 1 | 0 | 4 | 3 |
| 1 | 1 | 3 | 2 |

- Alta Entropia;
- Distribuição balanceada nas classes;
- **Preditores ruins**

**(b)**

**Figura 20: Caracterização de bons preditores utilizando ECM.**

**Fonte: Autoria própria.**

## 3.3 OTIMIZAÇÃO DA GRN INFERIDA

### 3.3.1 OTIMIZAÇÃO UTILIZANDO O ALGORITMO DE QUINE-MCCLUSKEY

Existem diversas formas de melhorar o processo de inferência utilizando hipótese sobre a estrutura da rede (LOPES et al., 2014), ou utilizando outros dados biológicos para aumentar o nível informação para aumentar a probabilidade da inferência (VICENTE et al., 2015). Entretanto, as redes podem não possuir tais características, ou não estarem disponíveis dados biológicos para a integração. Neste sentido é proposto neste trabalho a utilização do algoritmo de Quine-McCluskey (QM) (GSCHWIND; MCCLUSKEY, 1975) (veja Seção 2.3.3.2) como um método de pós-processamento das GRNs inferidas. O pós-processamento tem o objetivo de remover possíveis falsos genes preditores inferidos pelo método de seleção de características.

Esta abordagem permite minimizar as ligações entre os genes da rede de forma a remover genes inferidos irrelevantes. A vantagem desta abordagem é que não necessita de nenhuma hipótese *a priori*, nenhum dado adicional. O algoritmo de QM permite recuperar a expressão Booleana mínima dada uma tabela-verdade. Ou seja, caso alguma entrada não seja fundamental para a definição do circuito lógico este elemento é eliminado ao final do processo. De forma análoga as entradas do circuito podem ser representadas pelos estados dos genes preditores no tempo $t$ e a saída pode ser representa pelo estado do gene alvo no tempo $t+1$.

Assim foi proposto o uso do processo apresentado na Figura 21 para a seleção dos genes preditores mais relevantes para cada gene alvo da rede. Como entrada deste processo são necessários: os conjuntos dos genes preditores de cada gene alvo, os quais foram selecionados

a partir do método de seleção descrito na Seção 3.2 e os dados de expressão.

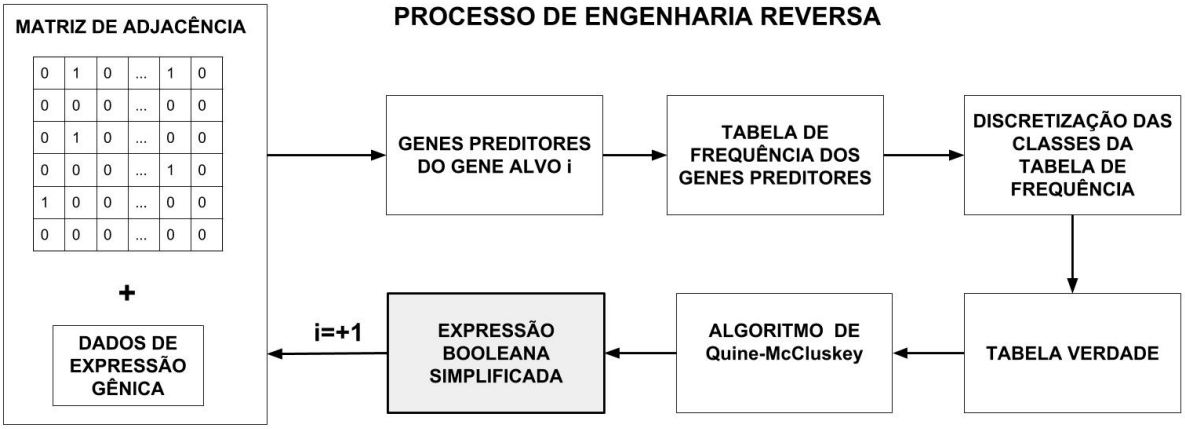



**Figura 21: Processo geral de recuperação das expressões Booleanas simplificadas dos genes preditores para cada gene alvo utilizando o Algoritmo de Quine-Mccluskey.**

**Fonte: Autoria própria.**

Para cada gene alvo ($Y_i$) é criada sua respectiva Tabela de Frequência (TF), onde para cada combinação dos estados dos genes preditores ($X_k$) no tempo $t$ é calculada a frequência do estado do gene alvo no tempo $t+1$, como é apresentado na Figura 22. Utilizando a codificação binária, os estados possíveis para 3 genes preditores são $2^N = 8$ (000, 001, 010, 011, 100, 110 e 111) e os estados possíveis para o gene alvo são 3 (zero, um e $\xi$).

TABELA DE FREQUÊNCIA

| t | | | t+1 | |
|---|---|---|---|---|
| | | | $Y_i$ | |
| $X_1$ | $X_2$ | $X_3$ | 0 | 1 |
| 0 | 0 | 0 | 144 | 50 |
| 0 | 0 | 1 | 20 | 144 |
| 0 | 1 | 0 | 144 | 20 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 144 | 30 |
| 1 | 0 | 1 | 10 | 9 |
| 1 | 1 | 0 | 50 | 244 |
| 1 | 1 | 1 | 47 | 244 |

TABELA VERDADE

| t | | | t+1 |
|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $Y_i$ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | ξ |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**Figura 22: Na Tabela de Frequência é apresentada a frequência de cada variação dos estados do conjunto $X$ (genes preditores), dada a variável $Y$ (gene alvo) nos estados ativos (1) e desativados (0). Posteriormente, as frequências dos estados são discretizadas para 0, 1 ou $\xi$ transformando em uma tabela verdade.**

**Fonte: Autoria própria.**

Com a TF montada é necessário discretizar as frequências em estados 0, 1 ou $\xi$ (*don't care*) para obter a tabela-verdade. A discretização neste trabalho utiliza a classe que prevalece com maior frequência, como no Figura 21 onde os estados 000 dos genes preditores $x_1$,$x_2$ e $x_3$

o gene alvo será discretizado para o estado 0, em razão do estado 0 possuir a maior frequência (144) comparado ao estado 1 (50). Neste trabalho o estado do gene alvo é discretizado para $\xi$ somente quando há empate entre as frequências dos estados 0 e 1.

Após obter a tabela-verdade do gene alvo *i* esta é submetida como entrada ao algoritmo de QM. No final do processo de QM é apresentada a respectiva expressão Booleana simplificada com os genes preditores mais relevantes para o gene alvo e suas respectivas interações representadas por operadores Booleanos (*AND*, *OR* e *NOT*). Posteriormente, é necessário remover da expressão os operadores Booleanos, o gene alvo e os genes preditores repetidos da expressão para obter apenas o subconjunto de genes preditores minimizado.

Neste trabalho foi utilizado o software *Simple Solver*[1] para resolver as tabelas-verdades, conforme apresentado na Figura 23. A razão da escolha deste software se deve ao fato de suportar uma lista de tabelas-verdades a partir de um protocolo simples, tal qual apresentado na Figura 23(a). Este fator é importante dado que o número de genes na rede é proporcional ao número de tabelas-verdades geradas que serão processadas, ou seja, para uma rede de 500 genes são construídos 500 tabelas verdades com números de preditores diferentes. A partir da entrada das *n* Tabelas Verdade, conforme apresentado na Figura 23(b), são obtidas *n* expressões Booleanas simplificadas, assim como apresentado na Figura 23(c). Por fim, é removido os genes preditores repetidos, os operadores Booleanas e o gene alvo mantendo apenas os genes preditores relevantes para a expressão. Os genes que restaram após este pós-processamento são mantidos na rede após este método de pós-processamento.

### 3.3.2 OTIMIZAÇÃO UTILIZANDO A REDE DE CONSENSO

O método de Rede de Consenso utiliza o resultado dos 3 algoritmos bioinspirados para inferir os genes preditores da rede, como mostrado na Figura 24. Neste método são considerados os genes preditores que estão contidos em pelo menos dois subconjuntos de preditores dos algoritmos.

Na Tabela 2 é apresentado um exemplo de uma inferência de três métodos (AM, ED, CAA), considerando o mesmo gene alvo. Neste exemplo o método utilizando o AM encontrou os genes preditores 91, 4 e 9. A abordagem utilizando o algoritmo ED encontrou os genes preditores 91 e 4. E, o método utilizando CAA encontrou o gene preditor 91. Neste contexto, a abordagem utilizando a rede de consenso mantém apena os genes 91, que está contido no subconjunto da inferência da abordagem utilizando AM, ED e CAA, e o gene preditor 4, que está contido no subconjunto das abordagens AM e ED. O gene 9 não é mantido na abordagem

[1] fonte acessada em 25 de Fevereiro de 2016: http://simplesolverlogic.com/index.html

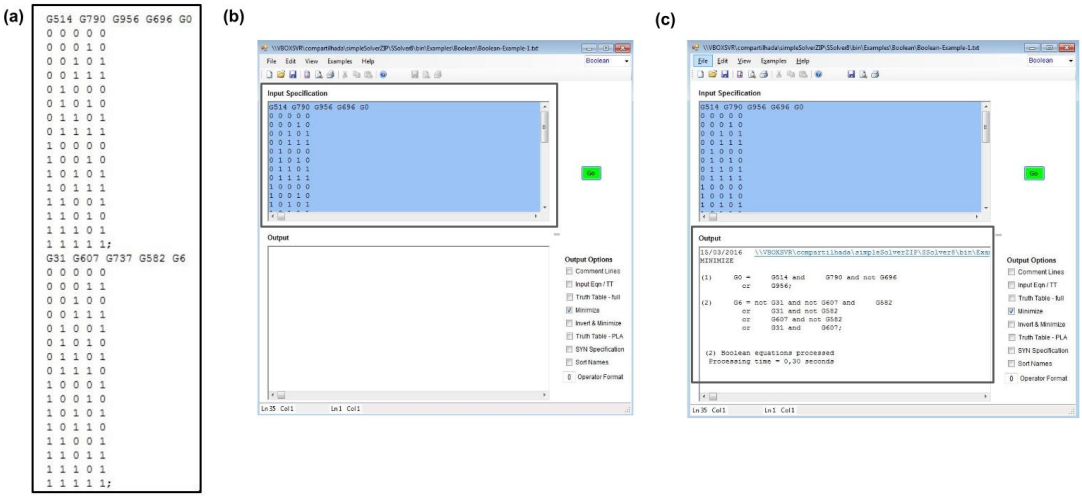



**Figura 23: Exemplo da etapa de simplificação das expressões Booleanas utilizando o software *Simples Solver*. (a) Protocolo utilizado pelo software (b) Entrada das $n$ tabelas verdade (c) Saída das $n$ expressões Booleanas simplificadas.**

**Fonte: Autoria própria.**

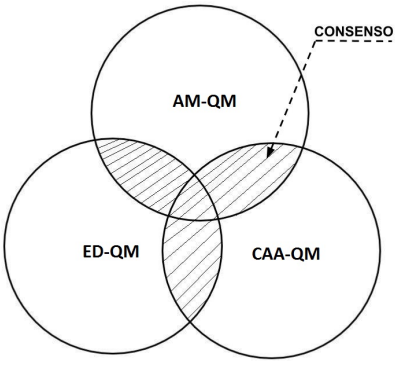



**Figura 24: Área hachurada representa os genes preditores considerados na rede de consenso e que foram inferidos por pelo menos dois dos algoritmos.**

da rede de consenso devido estar contido apenas no subconjunto da inferência da abordagem do AM. Este procedimento é feito em todos os genes preditores de cada alvo da rede.

**Tabela 2: Exemplo dos resultados das inferências pelos três algoritmos bioinspirados.**

| Método | Genes Preditores | | |
|---|---|---|---|
| **AM** | 91 | 4 | 9 |
| **ED** | 91 | 4 | - |
| **CAA** | 91 | - | - |
| **Rede de Consenso** | 91 | 4 | - |

**Fonte: Autoria própria.**

## 3.4 VALIDAÇÃO DO MÉTODO DE INFERÊNCIA

Após a etapa de identificação da rede é necessário avaliar a rede inferida a partir de alguma métrica. Tal qual apresentado na Seção 2.4 a rede gênica pode ser representada por uma matriz de adjacências $M$ que possui dimensão $n \times n$. Em razão desta matriz $M$ representar um grafo direcionado ela não é simétrica, ou seja, caso exista a ligação entre o gene $x_i$ e o $x_j$ na matriz é representada como $M(i, j) = 1 \neq M(i, j) = 0$.

Dougherty (2007) apresenta um método baseado na matriz de confusão para comparar a similaridade entre duas redes e consequentemente avaliar o método de inferência. Na matriz de confusão todos os acertos e erros das arestas inferidas são contabilizados. Na Tabela 3 são apresentados todos os elementos que compõem a tabela de confusão. Neste trabalho a matriz de confusão possui as seguintes semânticas que são contabilizadas: Verdadeiro Positivo (VP) representa a aresta que foi inferida e que também está contida na rede real, Falso Positivo (FP) representa uma conexão da rede inferida que não ocorre na rede real, Verdadeiro Negativo (VN) representa ligação que não foi inferida, mas que existe na rede real, e finalmente Falso Negativo (FN) que equivale a uma ligação que não foi inferida e que também não existe na rede real.

**Tabela 3: Matriz de Confusão com os Verdadeiro Positivo (VP), Falso Positivo (FP), Verdadeiro Negativo (VN) e Falso Negativo (FN).**

| Aresta \Conexão | Inferiu | não Inferiu |
|---|---|---|
| Contém | Verdadeiro Positivo (VP) | Falso Negativo (FN) |
| não Contém | Falso Positivo (FP) | Verdadeiro Negativo (VN) |

**Fonte: Adaptado de (CHOI et al., 2010)**

A partir da contabilização dos VP, FP, VN e FN na matriz de confusão é possível extrair algumas métricas, tais como a Precisão do método de inferência, como mostrado na Equação 13.

$$\text{Precisão} = \frac{VP}{VP+FP}, \tag{13}$$

onde $VP$ é o número de arestas inferidas e corretas, e $FP$ é o número arestas inferidas e que não existem na rede. Este método de avaliação de inferência é o mais comum e o mais utilizado. dado que a matriz é esparsa, ou seja, contém mais 0 do que 1 e a precisão, neste caso, é a mais interessante a ser identificada.

Outra medida que pode ser utilizada é o valor da sensibilidade, como mostrado na Equação 14.

$$Sensibilidade = \frac{VP}{VP+FN}, \tag{14}$$

onde $FN$ representa os genes que não foram inferidos, este valor representa a capacidade de

identificação dos genes preditores da rede, ou seja, quanto menos genes preditores o método identificar menor vai ser o valor de sensibilidade do método.

O valor de similaridade também é uma medida utilizada para comparar as GRNs e apresenta o balanço entre a precisão e a sensibilidade do método, e é omo mostrado na Equação 15.

$$Similaridade = \sqrt{\text{Precisão} * Sensibilidade}. \tag{15}$$

# 4 RESULTADOS E ANÁLISES

Para todos os experimentos apresentados neste trabalho foram utilizados computadores com processadores Quad Core, com Linux Ubuntu. Os algoritmos mostrados na Seção 4.1 foram implementados na linguagem Java e os algoritmos na Seção 4.2 foram implementados na linguagem ANSI-C.

## 4.1 COMPARAÇÃO ENTRE ALGORITMOS SEQUENCIAIS E BIOINSPIRADO

Nesta seção são comparados três algoritmos de busca: dois algoritmos sequenciais (o BSF e o BSFF) e um algoritmo de Computação Evolucionária, o algoritmo de Evolução Diferencial (EDD), proposto por Krause e Lopes (2013).

Para todos os testes foram utilizados as AGNs geradas pelo software jAGN (LOPES et al., 2011), os parâmetros para este experimento é apresentado na Tabela 4. Os seguintes parâmetros adotados foram: topologias dos modelos de rede complexas ER, WS e BA, com 50, 100 e 300 genes e variando o $\langle k \rangle$ de 1 a 3 e 300 tempos de expressão gênica. Para o algoritmo EDD cada experimento foi executado 10 vezes. Para todos os algoritmos de busca foram utilizadas a mesma função critério, a Entropia Condicional Média (MCE) baseada em (Martins Junior et al., 2008b) (veja Seção 2.7.1). Neste experimento os parâmetros dos algoritmos sequenciais foram definidos por Lopes et al. (2010) e para o EDD foram baseados em Krause e Lopes (2013) e a modelagem considerada foi um gene preditor por *bit* (veja Seção 3.2.1). A abordagem deste trabalho, tal qual apresentada por Jimenez et al. (2015), divide o problema de inferência de GRN em $n$ subproblemas de seleção de características totalizando 81000 execuções do método de seleção.

Os resultados do primeiro experimento são apresentados na Figura 25, o objetivo é comparar os métodos de busca aumentando a complexidade da rede em termos de quantidade de genes. Neste experimento são comparados os métodos BSF, BSFF e EDD utilizando a média da Precisão (PPV) (veja Seção 3.4) em porcentagem (%). Na média são consideradas todas as variações de topologia e de $\langle k \rangle$. Neste experimento foi possível avaliar a robustez dos métodos,

**Tabela 4: Parâmetros do experimentos.**

| Parâmetros | valores \modelos |
|---|---|
| **Tamanho da Rede** | $\{50, 100, 300\}$ |
| **Média de conexões** | $\{1, 2, 3\}$ |
| **Tamanho do sinal** | 300 |
| **Modelo Topológicos** | {ER, WS, BA } |
| **Modelo Funcional** | RBP |
| **Número de funções Booleanas** | 2 |
| **Probabilidade de funções Booleanas** | (96%, 4%) |
| **Algoritmos de Busca** | {BSF, BSFF, EDD } |
| **Função Critério** | ECM |

**Fonte: Autoria própria.**

mesmo aumentando a quantidade de genes a perda de desempenho foi baixa. O algoritmo BSF foi o que teve o pior desempenho considerando todas as variações do número de gene na rede, entretanto foi o algoritmo de que teve a menor variação de desempenho, com uma taxa de 2% de decréscimo. O algoritmo BSFF foi o segundo melhor método, atingindo 62% de precisão com 50 genes na rede e um decréscimo de 7% quando a rede aumenta para 300 genes. O algoritmo de EDD foi o melhor método considerando os 3 métodos avaliados e todas as variações de genes na rede, decrescendo 6% da menor rede para a maior rede. Comparando com o BSFF, o EDD teve um acréscimo de 7% de diferença e considerando o BSF a diferença foi de 19%.

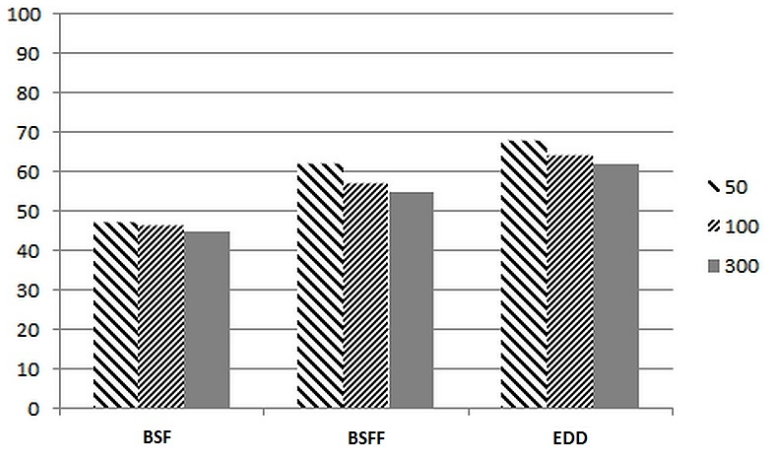



**Figura 25: Experimento considerando a variação na quantidade de genes na rede. A média do PPV do algoritmo BSF foi respectivamente 47%, 46% e 45%, BSFF apresentou 62%, 57% e 55%, e o algoritmo ED apresentou 68%, 64% e 62%.**

O segundo experimento desta seção apresenta o desempenho dos algoritmos de acordo com a topologia da rede. A Figura 26 apresenta a média de PPV em porcentagem (%) para todos os algoritmos sequenciais e o bioinspirado. De modo geral, a diferença entre as topologias não foi maior de que 2% para todos os algoritmos. Mais uma vez o algoritmo EDD proporcionou

melhores resultados quando comparado aos algoritmos BSF e BSFF. Como análise adicional, o algoritmo EDD apresentou um leve progresso para redes baseadas na topologia BA, apresentando melhor desempenho em redes que contêm *hubs*, diferentemente dos algoritmos BSF e BSFF.

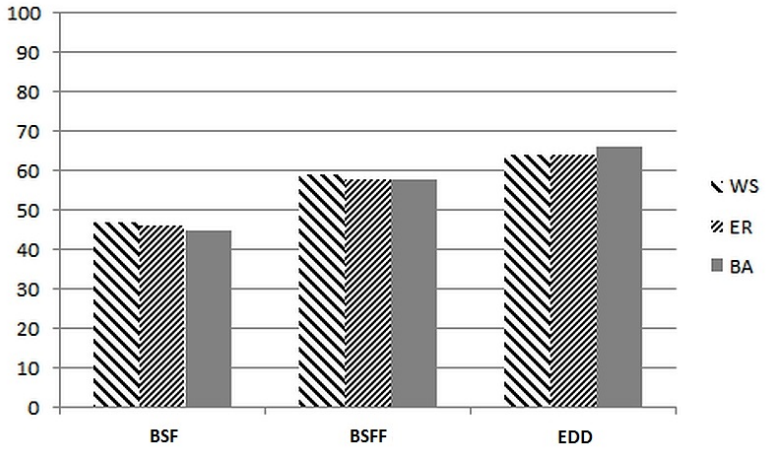



**Figura 26: Experimento dos algoritmos de acordo com as topologias WS, ER e BA. As médias do Precisão para o algoritmo BSF foram respectivamente 47%, 46% e 45%. O BSFF apresentou 59%, 58% e 58%, e o algoritmo EDD apresentou 64%, 64% e 66%.**

## 4.2 COMPARAÇÃO ENTRE MÉTODOS BIOINSPIRADOS

O segundo experimento teve o objetivo de comparar algoritmos meta-heurísticos da área de Computação Evolucionária (ED) e Inteligência de Enxames (CAA e AM). Tal qual no primeiro experimento, a inferência de GRNs é dividida em $n$ subproblemas de seleção de características. Para cada subproblema os algoritmos foram executados 20 vezes, e em cada iteração foi utilizada uma semente aleatória diferente. No total foram executados 6300 vezes cada algoritmo e a função *fitness* foi calculada $94,5 \times 10^9$ vezes no total .

Nesta abordagem todos os algoritmos utilizaram a mesma codificação com um número máximo de 4 genes preditores para cada indivíduo, como apresentado na Seção 3.2.1. Os parâmetros utilizados nos algoritmos foram os parâmetros padrões. Para o algoritmo ED a taxa de *crossover* ($CR = 0,8$) e a taxa de mutação ($F = 0,05$), AM a amplitude ($\alpha = 0,9$) e emissão de pulso ($\lambda = 0,9$), e CAA com número de abelhas trabalhadoras e oportunistas ($trabalhadora = 50$ e $oportunista = 50$) e a ativação das abelhas exploradoras ($limit = 100$), os parâmetros foram baseados no trabalho de Parpinelli et al. (2014). A codificação das soluções em todos os algoritmos utilizou valores na faixa $[-1 \ldots 1]$ e são discretizados com a função sigmóide, bem como no EDD apresentado no experimento anterior. Para uma comparação justa

foi adotado o mesmo número de indivíduos (100) e o mesmo número de gerações (50000) para todos os algoritmos.

A configuração neste experimento é apresentado na Tabela 5. O objetivo deste experimento foi a comparação entre os algoritmos bioinspirados AM, ED e CAA, nesta avaliação foi utilizada uma rede com topologia proposta por Babarasi-Albert (BA) contendo 1000 genes e 500 tempos de expressão gênica, utilizando como base características similares as GRNs inferidas em (MARBACH et al., 2012).

**Tabela 5: Parâmetros do experimentos.**

| **Parâmetros** | **valores \modelos** |
|---|---|
| **Tamanho da Rede** | {1000} |
| **Média de conexões** | {3} |
| **Tamanho do sinal** | 500 |
| **Modelo Topológicos** | {BA} |
| **Modelo Funcional** | RBP |
| **Número de funções Booleanas** | 2 |
| **Probabilidade de funções Booleanas** | (96%, 4%) |
| **Algoritmos de Busca** | {AM, ED, CAA } |
| **Função Critério** | ECM |

**Fonte: Autoria própria.**

Antes da execução dos algoritmos, foi feita uma análise sobre os dados temporais de expressão gênica com o intuito de identificar genes que possuem baixa variação ao longo do tempo. Os genes que possuem baixa variação de expressão ao longo do tempo não permitem identificar padrões na expressão e, consequentemente, não serão úteis para identificar seus respectivos preditores, logo podem ser desconsiderados do processo de inferência.

Na Figura 27 é apresentada a distribuição do número de vezes que os genes estavam expressos. Neste trabalho foram considerados apenas os genes que estariam expressos acima de 150 e abaixo de 350 vezes. No total foram considerados 350 genes como genes alvos estudados neste trabalho, os genes presentes nas colunas em preto no histograma da Figura 27.

Os resultados para os algoritmos AM, ED e CAA são apresentados na Figura 28. Neste *box plot* são apresentados os valores máximos, mínimos, medianas e quartis dos valores do ECM para cada algoritmo e considerando todas as inferências e as 20 repetições. Neste resultado é possível observar que todos os algoritmos tiveram uma grande variação nos 350 problemas de seleção de características. Entretanto, considerando a frequência de resultados o algoritmo AM foi o que obteve o pior resultado com os quartis e média com valores mais altos. O algoritmo CAA foi o segundo melhor e o algoritmo ED apresentou novamente menores valores de ECM, indicando que encontrava melhores preditores.

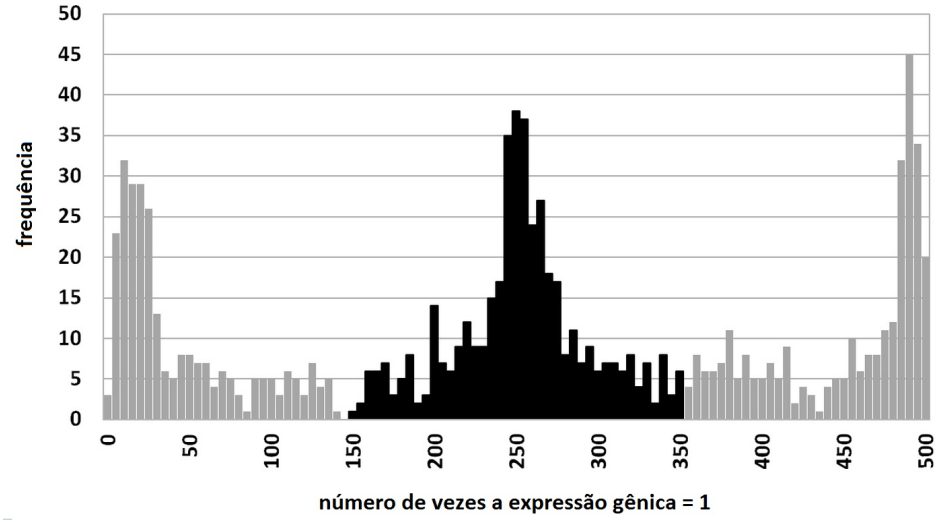



**Figura 27: Histograma da distribuição do número de vezes que os genes estão expressos. Os genes que estão nas extremidades tendem a estar, na maior parte do tempo, iguais a 1 ou 0. Estes não são apropriados para o processo de inferência de GRN.**

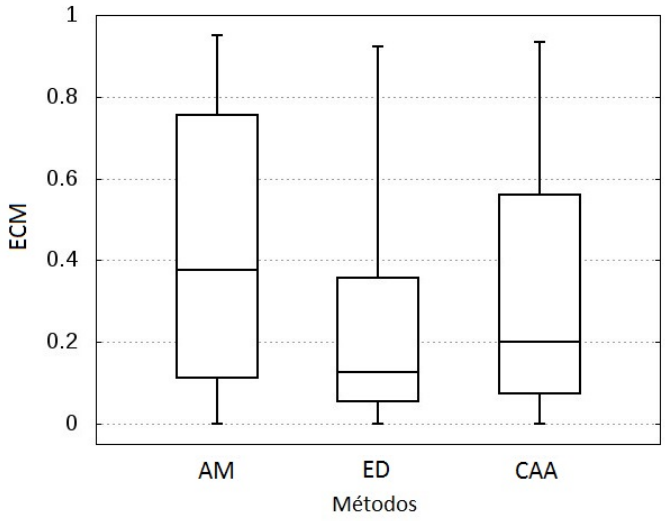



**Figura 28: Box plot do valor do ECM para os algoritmos AM, ED e CAA, neste gráfico foram considerados todas as repetições de experimentos e todos os genes alvos da rede.**

A segunda análise é referente à taxa de sucesso dos algoritmos, ou seja, o quanto os algoritmos conseguiram encontrar valores de ECM abaixo de $0,3$, este valor foi definido por Lopes et al. (2008). Na Figura 29 são apresentadas as taxas de sucesso. Nesta figura é possível observar duas barras para cada algoritmo. Na barra clara representa a proporção dos problemas que o algoritmo obteve sucesso. Neste caso o algoritmo ED obteve uma leve vantagem sobre o AM e CAA. Na barra escura são considerados apenas os problemas que possuam pelo menos um método com o ECM menor que $0,3$. Dentre estes problemas, o método que possuir o menor valor de ECM foi o que obteve sucesso, nesta análise o ED foi superior aos demais métodos.

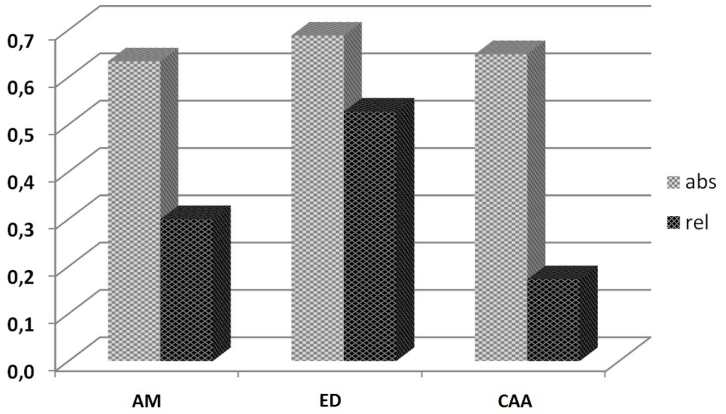



**Figura 29: Comparação entre os algoritmos AM, ED e CAA considerando a taxa de sucesso com um *threshold* de 0.3.**

A terceira análise é referente à comparação entre a rede com os melhores preditores inferidos por cada método bioinspirado e os genes da rede original, como apresentado na Figura 30. Para este teste foi considerado o melhor conjunto de genes preditores das 20 execuções para cada gene alvo, ou seja, foi o conjunto $X$ que possuía o menor valor ECM. Os melhores conjuntos $X$ de cada gene alvo foram comparados entre os algoritmos e também entre os genes da rede original, tal qual apresentado na Figura 30. Como medida para este teste foi utilizado a média do coeficiente de similaridade de Jaccard (JACCARD, 1912), de acordo com a Equação 16:

$$\bar{J}(A,B) = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i \cap B_i|}{|A_i \cup B_i|}, \tag{16}$$

onde $A_i$ e $B_i$ são os conjuntos de genes encontrados pelos métodos $A$ e $B$ para o *i-ésimo* gene alvo. A parte superior da equação representa o conjunto dos genes presentes tanto em A quanto em B. A parte inferior representa o conjunto de todos os genes distintos pertencentes a A ou B.

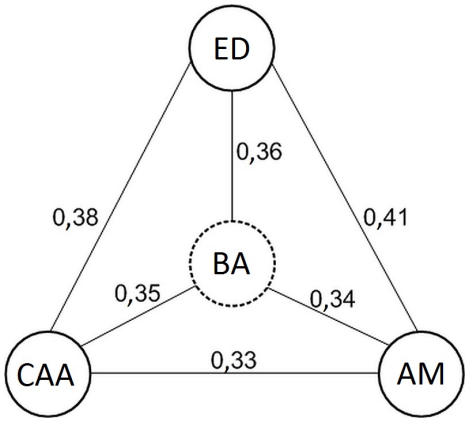



**Figura 30: Média do coeficiente de Jaccard entre os algoritmos AM, ED e CAA, e a topologia proposta por Barabasi-Albert (BA) original da rede.**

Este teste mostrou que todos os algoritmos obtiveram conjuntos de genes preditores

aproximadamente com a mesma similaridade com os genes da rede original. No entanto, as taxas de similaridade entre a rede original (representada na figura por BA) estão abaixo de ($\sim 0,35$), indicando que estes métodos foram capazes de encontrar parcialmente as soluções para inferência utilizando apenas os dados temporais de expressão gênica. Observando os métodos, o AM e o ED foram os algoritmos que tiveram o maior coeficiente de similaridade entre si. Neste experimento foi possível observar que o algoritmo ED apresentou a maior similaridade com a rede real, o qual foi refletido em todas as análises feitas neste experimento.

## 4.3 PÓS-PROCESSAMENTO COM O ALGORITMO DE QUINE-MCCLUSKEY E REDE DE CONSENSO

Neste experimento foram utilizados os dados obtidos através do segundo experimento (veja Seção 4.2), com os algoritmos bioinspirados (AM, ED, CAA) para inferência de GRNs utilizado dados temporais de expressão gênica. Os dados utilizados neste experimento são os genes preditores de cada gene alvo com o melhor *fitness* (obtido em todas as rodadas) de cada algoritmo bioinspirado. Os resultados da saída do último experimento passam por um pós-processamento utilizando o algoritmo Quine-McCluskey (veja Seção 2.3.3.2), a fim de realizar uma filtragem dos genes preditores de modo a eliminar os Falsos Positivos (FP). Também é apresentado os resultados uma rede de consenso que utiliza o conhecimento dos 3 algoritmos bioinspirados para inferir os genes preditores da rede, como mostrado na Seção 3.3.2.

Na Figura 31 são apresentadas as quantidades de genes preditores para cada algoritmo. Como é possível observar, para todos os algoritmos foi possível remover genes preditores sendo que para o algoritmo CAA a quantidade foi superior. Pode-se observar que a quantidade de genes removidos para o algoritmo AM e ED foram praticamente iguais, isso é refletido principalmente nas similaridades dos conjuntos de genes inferidos, conforme apresentado na Figura 30. As reduções de genes foram significativas sendo 364 genes preditores removidos (31%) para o algoritmo AM, 362 preditores ($31,75\%$) no algoritmo ED e 398 genes removidos (35%) para o algoritmo CAA.

O segundo teste apresenta o reflexo na qualidade das soluções considerando a remoção de genes preditores. Neste experimento a remoção de VP significa um baixo desempenho do método, pois remove genes que são genes preditores, por outro lado a remoção de FP é interessante, pelo fato de remover falsas inferências. Os resultados deste teste são apresentados na Figura 32, onde é possível observar que a quantidade de VP é baixa e a quantidade de FP é alta, o que mostra que o método de remoção por QM foi eficaz. Também é possível observar que o algoritmo ED teve os piores resultados neste experimento considerando a maior quantidade de

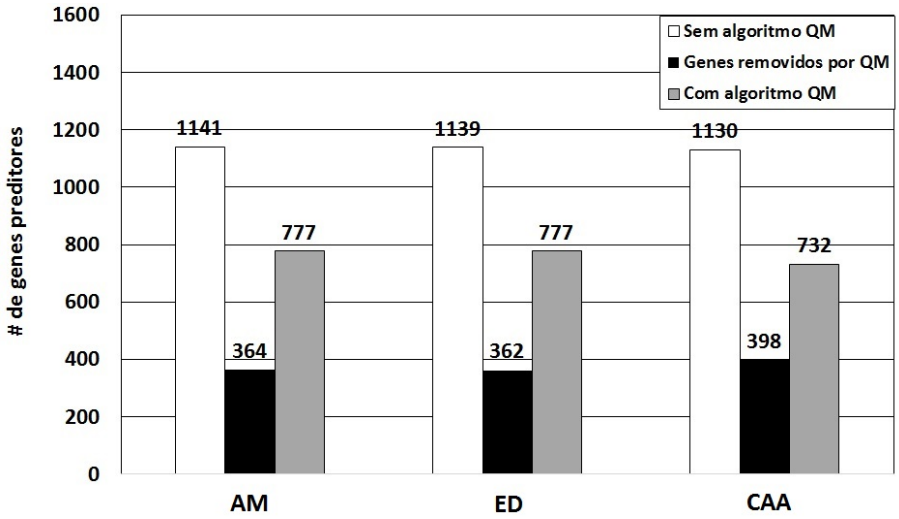



**Figura 31: Número de genes que foram removidos pelo algoritmo de Quine-McCluskey.**

remoções de VP (17) e o menor valor para FP (345). Em contrapartida o algoritmo CAA com o pós-processamento de QM consegue remover uma maior quantidade de genes FP (388) e com uma menor quantidade de VP (10).

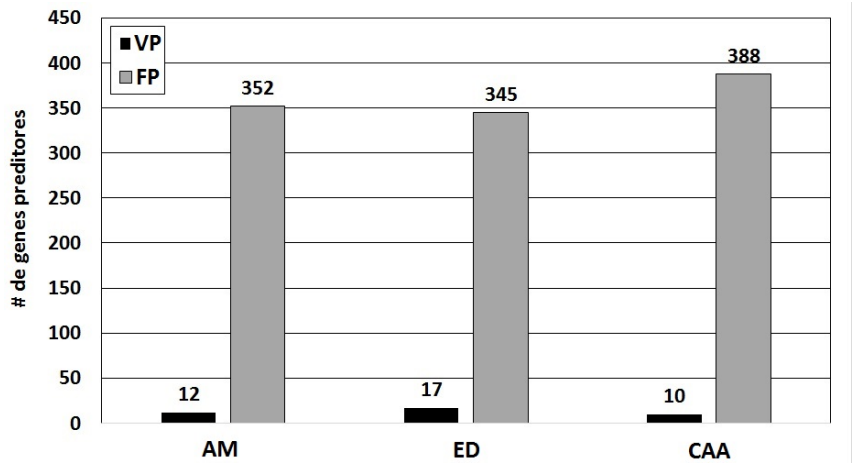



**Figura 32: Número de genes falsos positivos e verdadeiros positivos que foram removidos pelo algoritmo de Quine-McCluskey.**

Na Figura 33 são apresentados os métodos bioinspirados com a aplicação e sem a aplicação do pós-processamento do algoritmo de Quine-McCluskey (QM) e uma rede de consenso utilizando a votação dos métodos bioinspirados com o pós-processamento de QM. Neste teste é utilizada a média da precisão (veja Seção 3.4). Neste experimento é possível observar a variação da precisão dos métodos avaliados com e sem o pós-processamento. O algoritmo AM foi o que obteve o pior desempenho de precisão comparado aos outros métodos ($0,4181$) e com o pós-processamento o algoritmo também foi o pior quando comparado com os outros métodos ($0,5903$). O algoritmo ED obteve praticamente o mesmo resultado de precisão

(0,4381) comparado ao algoritmo CAA (0,4333) sem o pós-processamento. Entretanto, considerando o pós-processamento, assim como apresentado na Figura 32, o algoritmo CAA obteve melhores resultados pois houve remoção de mais genes FP e isto é refletido na precisão do método (0,654) , quando comparado ao ED (0,62). Considerando o método de consenso a precisão foi superior a todas as outras abordagens atingindo a precisão de (0,79).

Na Figura 33 considerando as médias de sensibilidade dos métodos. Neste teste não houve uma variação muito grande entre os métodos e entre os algoritmos com e sem pós-processamento ($\sim 0,47$). O algoritmo de ED foi o que teve uma ligeira queda de sensibilidade comparando o mesmo algoritmo com (0,47) e sem o pós-processamento por QM (0,49), em razão de ter sido o método que removeu o maior número de genes preditores VP. Na abordagem de consenso a sensibilidade não obteve nenhum aspecto diferente em relação aos outros métodos obtendo o valor médio de (0,47).

O último teste apresenta a comparação dos métodos bioinspirados considerando o balanço da precisão e da sensibilidade através do valor de similaridade, mostrado na Figura 33. Na figura é possível observar que os métodos com o pós-processamento obtiveram o melhor balanço entre precisão e sensibilidade em todos os casos, apresentado valores de similaridade acima de 0,5. Neste teste é possível observar que o método de consenso consegue balancear a média da precisão e manter a sensibilidade apresentando o melhor valor de similaridade (0,61) com a rede original.

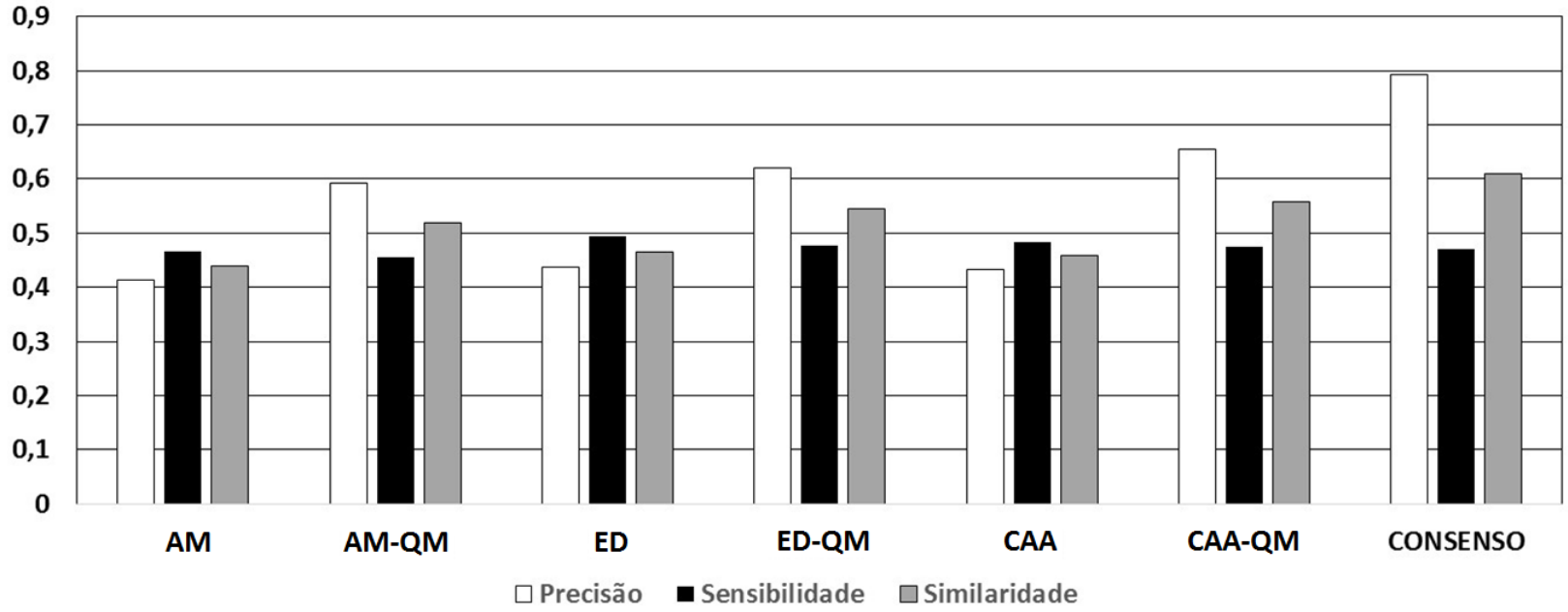



**Figura 33: Precisão, Sensibilidade e Similaridade dos algoritmos bioinspirados, com o algoritmo de Quine-McClusckey e a rede de consenso entre os algoritmos bioinspirados com o algoritmo de Quine-McClusckey.**

# 5 CONCLUSÃO

A inferência de GRNs ainda é um desafio em aberto na área de Bioinformática e, apesar de existirem muitos métodos propostos na literatura, as abordagens utilizando algoritmos bioinspirados ainda são pouco exploradas. Neste sentido este trabalho se concentrou na comparação de abordagens bioinspiradas para identificar as redes gênicas artificiais utilizando apenas dados temporais de expressão gênica.

Foram propostas abordagens de seleção de características utilizando algoritmos bioinspirados e sequenciais. A principal ideia foi dividir o problema de inferência em *n* subproblemas de seleção de características. Então, para cada subproblema de seleção foram obtidos os melhores preditores (características) para cada gene alvo da rede.

Neste sentido foram realizados três experimentos, onde o primeiro experimento compara os algoritmos sequenciais (BSF, BSFF) com um algoritmo de computação evolutiva (EDD). Foram utilizadas três topologias (ER, WS e BA), com o número de genes (100,200 e 300) e média de ligações entre os genes da rede (1, 2 e 3). O segundo experimento comparou os algoritmos de inteligência de enxames (AM e CAA) com um algoritmo de computação evolutiva (ED) utilizando uma rede gênica artificial baseada na topologia de Barabasi-Albert (BA), com 1000 genes e a média de ligações igual a 3.

As análises indicam que quanto mais genes estão na rede menor é a taxa de precisão de todos os algoritmos. Os experimentos mostraram que o EDD apresentou melhores resultados que os outros métodos, mesmo quando o número de genes era maior na rede e em todas as topologias. Como análise adicional foi percebido um pequeno decaimento de desempenho dos algoritmos BSF e BSFF quando inferidas redes com topologia BA, por outro lado o algoritmo EDD apresentou uma leve melhoria para esta topologia. O EDD apresentou melhores resultados que os outros algoritmos, e o algoritmo BSFF apresentou melhores resultados quando comparado ao BSF.

O segundo experimento teve o objetivo de comparar 3 abordagens de algoritmos bioinspirados para a inferência de GRNs utilizando uma Rede Artificial Gênica baseada em carac-

terísticas biológicas. As análises dos experimentos indicam que ambos os algoritmos apresentaram uma grande variabilidade nos resultados, entretanto é possível observar que nas frequências dos resultados possuem diferenças e que devem ser consideradas. O algoritmo DE apresentou os melhores desempenhos mostrando, com maior frequência, valores de ECM abaixo dos outros métodos. Quando investigada a taxa de sucesso dos algoritmos, o ED novamente obteve o melhor resultado. Comparando o grau de similaridade da topologia da rede real e a rede inferida pelos algoritmos bioinspirados o DE também atingiu o melhor índice de similaridade. No geral, em todas as análises o algoritmo DE apresentou os melhores resultados comparados aos algoritmos AM e CAA.

O terceiro experimento apresentou uma abordagem utilizando o algoritmo de Quine-McCluskey (QM) como uma etapa de pós-processamento após a inferência da GRN com o objetivo de otimizar a rede inferida. Os resultados deste experimento mostram que o método de QM consegue remover uma boa proporção de genes preditores FP em relação aos VP para os algoritmos AM, ED e CAA. A combinação do pós-processamento com o algoritmo CAA apresentou os melhores resultados considerando a precisão e a similaridade comparado aos outros métodos bioinspirados. O valor da sensibilidade não melhora utilizando o método de QM, mas consegue manter os resultados apresentando um bom custo-benefício. Também foi inferida uma rede de consenso que utiliza a votação dos 3 métodos bioinspirados com o processamento do QM. Esta abordagem apresentou uma precisão superior aos demais métodos. Já em relação à sensibilidade não teve uma expressiva mudança. Considerando a similaridade o método de consenso apresentou o melhor resultado.

Considerando GRNs cada vez maiores a serem inferidas, a tendencia é que meta heurísticas sejam cada vez mais difundidas e aplicadas ao processo de inferência. Neste sentido os métodos bioinspirados propostos neste trabalho aplicados ao processo de inferência se apresentaram adequados. A aplicação dos algoritmos bioinspirados na inferência ainda é pouco explorada e diversas modelagens para o problema ainda podem ser propostas, visando sanar esta deficiência foram propostas duas abordagens utilizando codificações contínuas, baseadas em (KRAUSE, 2014). Métodos de otimizações da GRN inferida também foram propostas neste trabalho como, a minimização das expressões Booleanas utilizando o método de QM e a utilização da rede de consenso das inferências dos métodos bioinspirados. Ambos os métodos de otimização promissores resultados permitindo inferências mais precisas.

## 5.1 TRABALHOS FUTUROS

A inferência de GRNs possibilita diversos aspectos podem ser trabalhos. Neste contexto as melhorias podem abranger aspectos do ajuste fino dos algoritmos apresentados neste trabalho, bem como propor outros algoritmos como AG (HOLLAND, 1992), OEP (KENNEDY; EBERHART, 1995), Algoritmo do Vagalume (YANG, 2010a), *parallel ECOlogically-inspired* (pECO) (BENITEZ et al., 2013), entre outros. Outras estratégias mais refinadas como o Algoritmo Memético (NERI; COTTA, 2012) podem ser uma solução interessante para tentar remover falsos preditores. Considerando o aspecto da função critério outras abordagens podem ser analisadas, como a Entropia de Tsallis (LOPES et al., 2011), Critério de Informação de Akaike (JIMENEZ et al., 2015), Incerteza Simétrica (EOM; ZHANG, 2004), entre outros. Visando aplicabilidade biológica dados da E. *coli*, S. *cerevisiae* (MARBACH et al., 2012) e A. *thaliana* (VICENTE et al., 2015) também podem ser utilizados para identificar o desempenho destes algoritmos.

# REFERÊNCIAS

AKUTSU, T. et al. Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. **Theoretical Computer Science**, v. 298, n. 1, p. 235 – 251, 2003.

BANATI, H.; BAJAJ, M. Firefly based feature selection approach. **International Journal of Computer Science Issues (IJCSI)**, v. 8, n. 4, p. 26–26, 2011.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **Science**, v. 286, n. 5439, p. 509–512, 1999.

BARRERA, J. et al. **Estimation of Probabilistic Genetic Networks of *Plasmodium falciparum* from Dynamical Expression Signals**. October 2005. 11–12 p. Poster session. First International Conference of the Brazilian Association for Bioinformatics and Computational Biology (X-Meeting).

BARRERA, J. et al. **Probabilistic Genetic Networks analysis of three *Plasmodium falciparum* strains from Dynamical Expression Signals**. 2006. Poster Session. 14th Annual International Conference On Intelligent Systems For Molecular Biology (ISMB).

BARRERA, J. et al. **Constructing probabilistic genetic networks of Plasmodium falciparum from dynamical expression signals of the intraerythrocytic development cycle**. New York, Philadelphia: Springer, 2007. 11–26 p.

BENITEZ, C.; PARPINELLI, R. S.; LOPES, H. S. A heterogeneous parallel ecologically-inspired approach applied to the 3d-AB off-lattice protein structure prediction problem. In: IEEE. **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC)**. Porto de Galinhas, Brazil, 2013. p. 592–597.

BISHOP, C. M. **Neural networks for pattern recognition**. Oxford, London: Oxford University press, 1995.

BOX, G. E. P. Evolutionary operation: A method for increasing industrial productivity. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, v. 6, n. 2, p. 81–101, 1957.

BREMERMANN, H. J. Optimization through evolution and recombination. In: **Proceedings of the Conference on Self-Organizing Systems – 1962**. Washington, DC: Spartan Books, 1962. p. 93–106.

CHAI, L. E. et al. A review on the computational approaches for gene regulatory network construction. **Computers in Biology and Medicine**, v. 48, p. 55–65, 2014.

CHARBONNIER, C.; CHIQUET, J.; AMBROISE, C. Weighted-LASSO for structured network inference from time course data. **Statistical Applications in Genetics and Molecular Biology**, v. 9, n. 1, p. 1–24, 2010.

CHICKERING, D. M.; HECKERMAN, D.; MEEK, C. Large-sample learning of Bayesian networks is NP-hard. **Journal of Machine Learning Research**, v. 5, n. 1, p. 1287–1330, dez. 2004.

CHOI, S.-S.; CHA, S.-H.; TAPPERT, C. C. A survey of binary similarity and distance measures. **Journal of Systemics, Cybernetics and Informatics**, v. 8, n. 1, p. 43–48, 2010.

CLAUSIUS, R. **The Mechanical Theory of Heat**. London: Macmillan, 1879.

CORDEIRO, J. A. **Meta-heurísticas aplicadas ao problema de projeção do preço de ações na bolsa de valores**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

COSTA, L. d. F. et al. Characterization of complex networks: A survey of measurements. **Advances in Physics**, v. 56, n. 1, p. 167–242, 2007.

CRICK, F. et al. Central dogma of molecular biology. **Nature**, v. 227, n. 5258, p. 561–563, 1970.

CRICK, F. H. C. On Protein Synthesis. **The Symposia of the Society for Experimental Biology**, v. 12, n. 1, p. 138–163, 1958.

DINGER, M. E. et al. Differentiating protein-coding and noncoding rna: Challenges and ambiguities. **PLOS Computational Biology**, v. 4, n. 11, p. 1–5, 11 2008.

DORIGO, M.; BIRATTARI, M.; STÜTZLE, T. Ant colony optimization. **IEEE Computational Intelligence Magazine**, v. 1, n. 4, p. 28–39, 2004.

DOUGHERTY, E. R. Validation of inference procedures for gene regulatory networks. **Current Genomics**, v. 8, n. 6, p. 351, 2007.

EOM, J.-H.; ZHANG, B.-T. Pubminer: machine learning–based text mining for biomedical information analysis. **Genomics & Informatics**, v. 2, n. 2, p. 99–106, 2004.

ERDÖS, P.; RÉNYI, A. On random graphs I. **Publicationes Mathematicae Debrecen**, v. 6, n. 1, p. 290–297, 1959.

ESTEVES, G. H. **Validação de procedimentos para medida de expressão gênica a partir de imagens de cDNA Microarray**. Dissertação (Mestrado) — Centro de Tratamento e Pesquisa do Hospital do Câncer, São Paulo, 2002.

FAITH, J. J. et al. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. **PLoS Biology**, v. 5, n. 1, p. 8–16, 2007.

FLOYD, T. L. **Sistemas digitais: fundamentos e aplicações**. Porto Alegre: Bookman, 2007.

FOGEL, L. J. Autonomous automata. **Industrial Research**, v. 4, n. 2, p. 14–19, 1962.

FRIEDBERG, R. M. A learning machine: Part I. **IBM Journal of Research and Development**, v. 2, n. 1, p. 2–13, 1958.

GALLO, C. A.; CARBALLIDO, J. A.; PONZONI, I. Inference of gene regulatory networks based on association rules. In: ____. **Biological Knowledge Discovery Handbook**. Nova Jersey: John Wiley & Sons, 2013. p. 803–838.

GARRO, B. A.; RODRÍGUEZ, K.; VÁZQUEZ, R. A. Classification of dna microarrays using artificial neural networks and abc algorithm. **Applied Soft Computing**, v. 38, n. 1, p. 548–560, 2016.

GONÇALVES, B. Projeto de circuitos lógicos. Apresentação de Projeto de Circuitos Lógicos. 2008.

GSCHWIND, H. W.; MCCLUSKEY, E. J. **Design of Digital Computers**. New York, USA: Springer–Verlag, 1975.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **The Journal of Machine Learning Research**, v. 3, n. 1, p. 1157–1182, 2003.

HAMMING, R. W. Error detecting and error correcting codes. **Bell System Technical Journal**, v. 29, n. 2, p. 147–160, 1950.

HATTORI, L. T.; LOPES, H. S.; LOPES, F. M. A discretized differential evolution algorithm for the inference of gene regulatory networks. In: IEEE. **2015 Latin America Congress on Computational Intelligence (LA-CCI)**. Curitiba, 2015. p. 1–6.

HECKER, M. et al. Gene regulatory network inference: data integration in dynamic models - a review. **Biosystems**, v. 96, n. 1, p. 86–103, 2009.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1992.

HOSSINI, A. M. et al. Induced pluripotent stem cell-derived neuronal cells from a sporadic Alzheimer's disease donor as a model for investigating ad-associated gene regulatory networks. **BMC Genomics**, v. 16, n. 1, p. 84–88, 2015.

HUANG, C.-Y.; SUN, C.-T.; LIN, H.-C. Influence of local information on social simulations in small-world network models. **Journal of Artificial Societies and Social Simulation**, v. 8, n. 4, p. 8–17, 2005.

HUYNH-THU, V. A. et al. Inferring regulatory networks from expression data using tree-based methods. **PloS One**, v. 5, n. 9, p. 127–135, 2010.

IBA, H.; MIMURA, A. Inference of a gene regulatory network by means of interactive evolutionary computing. **Information Sciences**, v. 145, n. 3, p. 225–236, 2002.

JACCARD, P. The distribution of the flora in the alpine zone. **New Phytologist**, v. 11, n. 2, p. 37–50, 1912.

JIMENEZ, R. D.; Martins Jr., D. C.; SANTOS, C. S. One genetic algorithm per gene to infer gene networks from expression data. **Network Modeling Analysis in Health Informatics and Bioinformatics**, v. 4, n. 1, p. 1–22, 2015.

KARABOGA, D.; BASTURK, B. On the performance of artificial bee colony (ABC) algorithm. **Applied Soft Computing**, v. 8, n. 1, p. 687–697, 2008.

KAUFFMAN, S. A. Metabolic stability and epigenesis in randomly constructed genetic nets. **Journal of Theoretical Biology**, v. 22, n. 3, p. 437–467, 1969.

KAUFFMAN, S. A. **The origins of order: Self organization and selection in evolution**. Oxford: Oxford University Press, 1993.

KAUFFMAN, S. A. **The origins of order: Self-organization and selection in evolution**. Oxford: Oxford University Press, 1993.

KELEMEN, A.; ABRAHAM, A.; CHEN, Y. **Computational intelligence in bioinformatics**. Berlin Heidelberg: Springer, 2008.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Neural Networks, 1995. Proceedings., IEEE International Conference on**. [S.l.: s.n.], 1995. v. 4, p. 1942–1948.

KOZA, J. R. **Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems**. Stanford, CA, USA, 1990.

KRAUSE, J. **Programação matemática e evolução diferencial para a otimização de redes de dutos**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, 2014.

KRAUSE, J.; CORDEIRO, J. A.; LOPES, H. S. Comparação de métodos de computação evolucionária para o problema da mochila multidimensional. In: LOPES, H. S.; RODRIGUES, L. C. de A.; STEINER, M. T. A. (Ed.). **Meta-Heurísticas em Pesquisa Operacional**. 1. ed. Curitiba, PR: Omnipax, 2013. cap. 6, p. 87–98.

KRAUSE, J. et al. A survey of swarm algorithms applied to discrete optimization problems. In: YANG, X.-S. et al. (Ed.). **Swarm Intelligence and Bio-Inspired Computation: Theory and Applications**. Amsterdam, The Netherlands: Elsevier Science, 2013. p. 169–192.

KRAUSE, J.; LOPES, H. A comparison of differential evolution algorithm with binary and continuous encoding for the MKP. In: BRICS-CCI. **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. Recife, PE: IEEE Press, 2013. p. 381–387.

LEHNINGER, A. L. L. **Princípios de bioquímica**. São Paulo: Sarvier, 1989.

LIANG, S.; FUHRMAN, S.; SOMOGYI, R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. **Journal of Theoretical Biology**, v. 2, n. 1, p. 1–12, 1998.

LOPES, F.; OLIVEIRA, E. A. de; CESAR, R. M. Inference of gene regulatory networks from time series by tsallis entropy. **BMC Systems Biology**, v. 5, n. 1, p. 1–13, 2011.

LOPES, F. M. **Redes complexas de expressão gênica: síntese, identificação, análise e aplicações**. Tese (Doutorado) — Universidade de São Paulo, 1 2011.

LOPES, F. M.; CESAR, R. M.; COSTA, L. F. AGN Simulation and Validation Model. In: THIRD BRAZILIAN SYMPOSIUM ON BIOINFORMATICS. **Advances in Bioinformatics and Computational Biology**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 169–173.

LOPES, F. M. et al. A feature selection technique for inference of graphs from their known topological properties: Revealing scale-free gene regulatory networks. **Information Sciences**, v. 3, n. 4, p. 1–15, 2014.

LOPES, F. M.; JR, R. M. C.; COSTA, L. D. F. Gene expression complex networks: synthesis, identification, and analysis. **Journal of Computational Biology**, v. 18, n. 10, p. 1353–1367, 2011.

LOPES, F. M. et al. SFFS-MR: A floating search strategy for grns inference. In: PRIB. **Pattern Recognition in Bioinformatics: 5th IAPR International Conference**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 407–418.

LOPES, F. M.; MARTINS, D. C.; CESAR, R. M. Feature selection environment for genomic applications. **BMC Bioinformatics**, v. 9, n. 1, p. 451–459, 2008.

MADHAMSHETTIWAR, P. B. et al. Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. **Genome medicine**, v. 4, n. 5, p. 1–16, 2012.

MANDAL, S.; KHAN, A. Reverse engineering of gene regulatory networks based on s-systems and bat algorithm. **Journal of Bioinformatics and Computational Biology**, v. 1, n. 2, p. 22–44, 2016.

MARBACH, D. et al. Wisdom of crowds for robust gene network inference. **Nature Methods**, v. 9, n. 8, p. 796–804, 2012.

MARGOLIN, A. A. et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. **BMC Bioinformatics**, v. 7, n. 1, p. 7–17, 2006.

MARILL, T.; GREEN, D. M. On the effectiveness of receptors in recognition systems. **IEEE Transactions on Information Theory**, v. 9, n. 1, p. 11–17, 1963.

MARTINS, D. C. et al. Network properties of intrinsically multivariate predictive genes. In: GENSIPS. **International Workshop on Genomic Signal Processing and Statistics.** Phoenix-AZ, 2008. p. 1–2.

Martins Junior, D. C. **Seleção de características e predição intrinsecamente multivariada em identificação de redes de regulação gênica**. Tese (Doutorado) — Universidade de São Paulo, Jan 2009.

Martins Junior, D. C. et al. Intrinsically multivariate predictive genes. **IEEE Journal of Selected Topics in Signal Processing**, v. 2, n. 3, p. 424–439, June 2008.

Martins Junior, D. C. et al. Intrinsically multivariate predictive genes. **IEEE Journal of Selected Topics in Signal Processing**, v. 2, n. 3, p. 424–439, 2008.

MEYER, P. E. et al. Information-theoretic inference of large transcriptional regulatory networks. **EURASIP journal on bioinformatics and systems biology**, v. 1, n. 1, p. 1–9, 2007.

MILGRAM, S. The small world problem. **Psychology Today**, v. 2, n. 1, p. 60–67, 1967.

NERI, F.; COTTA, C. Memetic algorithms and memetic computing optimization: A literature review. **Swarm and Evolutionary Computation**, v. 2, n. 1, p. 1–4, 2012.

PALIT, S. et al. A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm. In: IEEE. **International Conference on Computer and Communication Technology (ICCCT)**. India, 2011. v. 2, p. 428–432.

PARPINELLI, R. S. et al. Performance analysis of swarm intelligence algorithms for the 3D-AB off-lattice protein folding problem. **Journal of Multiple-Valued Logic and Soft Computing**, v. 22, n. 1, p. 267–286, 2014.

PUDIL, P.; NOVOVIČOVÁ, J.; KITTLER, J. Floating search methods in feature selection. **Pattern Recognition Letters**, v. 15, n. 11, p. 1119–1125, 1994.

QUINE, W. V. The problem of simplifying truth functions. **American Mathematical Monthly**, v. 59, n. 8, p. 521–531, 1952.

QUINE, W. V. A way to simplify truth functions. **American Mathematical Monthly**, v. 62, n. 9, p. 627–631, 1955.

RECHENBERG, I. Cybernetic solution path of an experimental problem. In: **Royal Aircraft Establishment Translation**. Farnborough: Royal Aircraft Establishment Ministry of Aviation, 1965.

RUBIO-LARGO, Á.; VEGA-RODRÍGUEZ, M. A.; GONZÁLEZ-ÁLVAREZ, D. L. Hybrid multiobjective artificial bee colony for multiple sequence alignment. **Applied Soft Computing**, v. 41, n. 1, p. 157–168, 2016.

SAKAMOTO, E.; IBA, H. Inferring a system of differential equations for a gene regulatory network by using genetic programming. In: IEEE. **Proceedings of the Congress on Evolutionary Computation**. Seoul, 2001. v. 1, p. 720–726.

SCHAFFTER, T.; MARBACH, D.; FLOREANO, D. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. **Bioinformatics**, v. 27, n. 16, p. 2263–2270, 2011.

SHALON, D.; SMITH, S. J.; BROWN, P. O. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. **Genome Research**, v. 6, n. 7, p. 639–645, 1996.

SHANNON, C. E. A mathematical theory of communication. **SIGMOBILE Mobile Computing and Communications Review**, v. 5, n. 1, p. 3–55, jan. 1963.

SHMULEVICH, I.; DOUGHERTY, E. R. **Genomic Signal Processing (Princeton Series in Applied Mathematics)**. Princeton, NJ, USA: Princeton University Press, 2007.

SIMÕES, R. de M. et al. Urothelial cancer gene regulatory networks inferred from large-scale rnaseq, bead and oligo gene expression data. **BMC Systems Biology**, v. 9, n. 1, p. 21, 2015.

SOMOL, P.; PUDIL, P.; KITTLER, J. Fast branch & bound algorithms for optimal feature selection. **IEEE Transactions on Pattern Analysis and Machine IntelligenceS**, v. 26, n. 7, p. 900–912, 2004.

STORN, R.; PRICE, K. **Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces**. Berkeley: ICSI Berkeley, 1995.

TERFVE, C. et al. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. **BMC Systems Biology**, v. 6, n. 1, p. 133, 2012.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas digitais: princípios e aplicações**. São Paulo - SP: Pearson Education, 2003.

VELCULESCU, V. E. et al. Serial analysis of gene expression. **Science**, v. 270, n. 5235, p. 484–487, 1995.

VICENTE, F. F. R. et al. A feature selection approach for evaluate the inference of GRNs through biological data integration – A case study on A. Thaliana. In: CIARP. **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 20th Iberoamerican Congress**. Cham: Springer International Publishing, 2015. p. 667–675.

VOET, E.; VOET, J.; PRATT, C. W. Fundamentals of biochemistry: Life at the molecular level. **Biochemistry and Molecular Biology Education**, v. 36, n. 4, p. 319–320, 2008.

WANG, Z.; GERSTEIN, M.; SNYDER, M. Rna-seq: a revolutionary tool for transcriptomics. **Nature Reviews Genetics**, v. 10, n. 1, p. 57–63, 2009.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. **Nature**, v. 393, n. 6684, p. 440–442, 1998.

WHITNEY, A. W. A direct method of nonparametric measurement selection. **IEEE Transactions on Computers**, v. 20, n. 9, p. 1100–1103, 1971.

XU, R.; VENAYAGAMOORTHY, G. K.; WUNSCH, D. C. Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. **Neural Networks**, Elsevier, v. 20, n. 8, p. 917–927, 2007.

YANG, X.-S. **Nature-inspired metaheuristic algorithms**. RICHMOND, TX, USA: Luniver press, 2010.

YANG, X.-S. **A New Metaheuristic Bat-Inspired Algorithm**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 65–74 p.

ZHANG, B. et al. Characterization of genetic networks associated with Alzheimer's disease. In: **Systems Biology of Alzheimer's Disease**. New York, NY: Springer New York, 2015. p. 459–477.

# ANEXO A – ALGORITMOS BIOINSPIRADOS

**Algoritmo 1:** Pseudo-código do algoritmo ED, de acordo com (KRAUSE; LOPES, 2013).

função ED($NP, CR, F, NV$);
**Gerar** aleatoriamente a população inicial($NP$ indivíduos);
$\chi \leftarrow random(NP, NV)$;
**Computar** o *fitness* para todos os indivíduos da população;
$fit_{\chi} \leftarrow f(\chi)$;
**while** *critério de parada=FALSO* **do**
    **for** $i = 1$ **to** $NP$ **do**
        $\vec{v}_i^{G+1} \leftarrow$ mutação$(\vec{\chi}_i^G, F)$;
        $\vec{u}_i^{G+1} \leftarrow crossover(\vec{\chi}_i^G, \vec{v}_i^{G+1}, CR)$;
    $fit_u \leftarrow f(\vec{u})$;
    **for** $i = 1$ **to** $NP$ **do**
        **if** $fit_u(i) > fit_x(i)$ **then**
            $\vec{\chi}_i^{G+1} \leftarrow \vec{u}_i^{G+1}$;
        **else**
            $\vec{\chi}_i^{G+1} \leftarrow \vec{\chi}_i^G$;
    **Atualiza** *critério de parada*;

**Algoritmo 2:** Pseudo-código do algoritmo CAA, de acordo com (PARPINELLI et al., 2014)

função CAA($NP, NV, limit$) ;
**Gerar** aleatoriamente a população inicial($NP$ abelhas);
$\chi \leftarrow random(NP, NV)$;
$fit_x \leftarrow avaliar(x)$;
count $= 0$;
**while** *critério de parada=FALSO* **do**
    **Fase: abelhas empregadas**
    **for** $i \leftarrow 1$ **to** $NP/2$ **do**
        Selecionar $k$ e $j$, $k \in \{1, NP\}$, $j \in \{1, d\}$ ;
        $\vec{u}_i^{G+1} = \vec{x}_{ij}^G + rand * (\vec{\chi}_{ij}^G - \vec{\chi}_{kj}^G)$, $rand \in [0,1]$;
        **if** $fit_u$ *é menor que* $fit_x$ **then**
            $\vec{\chi}_i^G = \vec{u}_i^{G+1}$;
        **else**
            $count = count + 1$;
    **Fase: abelha oportunista**
    **for** $i \leftarrow NP/2$ **to** $NP$ **do**
        Aplicar o processo de seleção por Roleta
        $P(\vec{\chi}_k) = \frac{fit_k}{\sum NPk=if(\vec{\chi}_k)}$
        Produz $\vec{u}_i^{G+1}$
        **if** $fit_u$ *é menor que* $fit_\chi$ **then**
            $\vec{x}_i^G = \vec{u}_i^{G+1}$;
        **else**
            $count = count + 1$;
    **Fase: abelha exploradora**
    **if** *count é maior que limit* **then**
        $\vec{\chi}_i^{G+1} = \vec{\chi}_i^G random$;
        $count = 0$;
    **Atualiza** *critério de parada*;

**Algoritmo 3:** Pseudo-código do AM, de acordo com (PARPINELLI et al., 2014).

função AM($NP, NV, \alpha, \lambda$) ;
Define a frequência $f_i$ de cada $\chi_i$ ;
Inicializar a taxa de pulso $pr_i$ e amplitude $A_i$;
**Gerar** aleatóriamente a população inicial ($NP$ morcegos);
$\chi \leftarrow random(NP, NV)$;
$fit_\chi \leftarrow avaliar(\chi)$;
Ordenar os morcegos pelo valor de *fitness* e armazenar o melhor morcego $x_*$;
**while** *critério de parada=FALSO* **do**
    **for** $i \leftarrow 1$ **to** $NP$ **do**
        $f_i = f_{min} + (f_{max} - f_{min})\beta$, $\beta \in [0,1]$;
        $\vec{vel}_i^{G+1} = \vec{vel}_i^{G} + (\vec{vel}_i^{G} - \vec{vel}_*^{G})$;
        $\vec{u}_i^{G+1} = \vec{\chi}_i^{G} + \vec{vel}_i^{G+1}$;
        **if** $rand > pr_i$ **then**
            $\vec{u}_i^{G+1} = \chi_* + \varepsilon\bar{A}$, $\varepsilon \in [-1,1]$;
        Gerar uma nova solução perturbando na dimensão do $\vec{u}_i^{G+1}$.;
        **if** $rand < A_i$ & $fit_{\chi_i} < fit_{\chi_*}$, $rand \in [0,1]$ **then**
            $\vec{\chi}_i^{G+1} = \vec{u}_i^{G+1}$;
            $pr_i^{G+1} = 1 - exp(-\lambda t)$;
            $A_i^{G+1} = \alpha A_i^{G}$;
    atualizar o melhor morcego $\chi_*$;
    **Atualizar** *critério de parada*;