UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

JADER MAIKOL CALDONAZZO GARBELINI

Abordagem baseada em algoritmos meméticos para descoberta de motivos biológicos

DISSERTAÇÃO-MESTRADO

CORNÉLIO PROCÓPIO

2017

JADER MAIKOL CALDONAZZO GARBELINI

Abordagem baseada em algoritmos meméticos para descoberta de motivos biológicos

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Bioinformática (PPGBIOINFO) da Universidade Tecnológica Federal do Paraná como parte dos requisitos necessários para a obtenção do grau de Mestre em Bioinformática.

Orientador: Prof. Dr. Danilo Sipoli Sanches

Coorientador: Prof. Dr. André Yoshiaki Kashiwabara

CORNÉLIO PROCÓPIO

2017

Dados Internacionais de Catalogação na Publicação

G213 Garbelini, Jader Maikol Caldonazzo
Abordagem baseada em algoritmos meméticos para descoberta de motivos biológicos / Jader Maikol Caldonazzo Garbelini. – 2017.
103 f. : il. color. ; 31 cm

Orientador: Danilo Sipoli Sanches.
Coorientador: André Yoshiaki Kashiwabara.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. Cornélio Procópio, 2017.
Bibliografia: p. 94-103.

1. Computação evolutiva. 2. Heurística. 3. Memética. 4. Inteligência computacional. 5. Bioinformática – Dissertações. I. Sanches, Danilo Sipoli, orient. II. Kashiwabara, André Yoshiaki, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. IV. Título.

CDD (22. ed.) 572.80285

Biblioteca da UTFPR - Câmpus Cornélio Procópio

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Programa de Pós-Graduação em Bioinformática

Título da Dissertação Nº 01:

"ABORDAGEM BASEADA EM ALGORITMOS MEMÉTICOS PARA DESCOBERTA DE MOTIVOS BIOLÓGICOS".

por

Jader Maikol Caldonazzo Garbelini

Orientador: **Prof. Dr. Danilo Sipoli Sanches**
Coorientador: **Prof. Dr. André Yoshiaki Kashiwabara**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM BIOINFORMÁTICA – Linha de Pesquisa: Biologia Computacional e Sistêmica, pelo Programa de Pós-Graduação em Bioinformática – PPGBIOINFO – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 09h do dia 06 de março de 2017. O trabalho foi _____ pela Banca Examinadora, composta pelos professores:

_____
Prof. Dr. Danilo Sipoli Sanches
(Presidente)

_____
Prof. Dr. Laurival Antonio Vilas
(UEL-PR)

_____
Prof. Dr. Renato Tinós
(USP-SP)

Visto da coordenação: _____
Fabricio Martins Lopes
Coordenador do Programa de Pós-Graduação em Bioinformática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico este trabalho a Deus que me inspirou e a minha família que me apoiou.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar ideias, saúde e forças para superar as dificuldades. À Universidade Tecnológica Federal do Paraná pelo ambiente criativo e amigável que proporciona. Ao meu orientador Prof. Dr. Danilo Sipoli Sanches, pelas orientações e pelo empenho dedicado à elaboração deste trabalho. Ao meu coorientador Prof. Dr. André Yoshiaki Kashiwabara pela orientação, apoio e confiança. A Dra. Fernanda Moraes pelos direcionamentos e conselhos nos conceitos biológicos. Aos amigos Erinaldo, Cynara, Tatiane, Juliana e Fábio pela sincera amizade, união e momentos de estudo. Ao Prof. Dr. Fabrício Martins Lopes e Prof. Dr. Alexandre Rossi Paschoal pelas aulas e ensinamentos que foram de fundamental importância. A Profa. Dra. Francismar Corrêa Marcelino-Guimarães e ao Prof. Dr. Laurival Antonio Vilas-Boas pelo compartilhamento de importantíssimos conhecimentos biológicos. A CAPES e a UTFPR-CP pelo suporte financeiro e por incentivar a ciência e tecnologia. Aos meus pais que sempre me incentivaram a estudar. Um agradecimento especial a minha esposa Rose, pela paciência, ajuda e dedicação. A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

"O que sabemos é uma gota; o que ignoramos é um oceano."
(Isaac Newton)

RESUMO

A localização dos Sítios de Ligação dos Fatores de Transcrição (TFBS, do inglês *Transcription Factor Binding Sites*) é considerado um dos principais desafios da Bioinformática. A sua correta identificação desempenha um papel importante na compreensão dos mecanismos de regulação gênica e desenvolvimento de novas drogas. A descoberta de *motivos de novo* é uma tarefa difícil e a construção de programas computacionalmente eficazes é necessária para melhorar a compreensão e o estudo dos transcritos celulares. Isso permite apontar e eleger elementos recorrentes em um conjunto de sequências para posterior investigação biológica, tais como os resultados de experiências de expressão diferencial de elevado desempenho. Neste trabalho apresentamos o Arcabouço Memetico para Descoberta de *Motivos* (MFMD, do inglês *Memetic Framework for Motif Discovery*), um algoritmo cuja construção foi inspirada na teoria dos memes e utilizou como base duas heurísticas – uma construtiva semi-gulosa baseada no GRASP e outra baseada no VNS – bem como um otimizador global baseado nos algoritmos evolutivos. Quando avaliado em conjuntos de dados sintéticos e reais, o MFMD superou as principais ferramentas de detecção de *motivos* existentes. Essa nova abordagem foi comparada à outras técnicas bem conhecidas da literatura e os resultados sugerem uma melhora significativa nas medidas de desempenho alcançadas pelo MFMD em relação aos algoritmos confrontados.

Palavras-chave: *motivos*, algoritmos evolutivos, algoritmos memeticos, heurísticas, sítios de ligação dos fatores de transcrição

ABSTRACT

The location of Transcription Factor Binding Sites (TFBS) is considered one of the main problems of Bioinformatics. Their correct identification plays an important role in understanding the mechanisms of genetic regulation and development of new drugs. The de novo motif discovery is a difficult task and the construction of computationally effective programs is necessary to improve the understanding and study of cell transcripts. This allowed to point and choose recurring elements in a set of sequences for further biological investigation, such as the results of high performance differential expression experiments. In this work we present the Memetic Framework for Motif Discovery (MFMD), an algorithm whose construction was inspired by the theory of memes and based on two heuristics - a semi-greedy construct based on GRASP and another based on VNS - as well as a global optimizer based on the evolutionary algorithms. When evaluated in synthetic and real datasets, MFMD has outperformed the main existing motif detection tools. This new approach was compared to other techniques well known in the literature and the results suggested a significant improvement in the performance measures achieved by MFMD in relation to the algorithms faced.

Keywords: motifs, evolutionary algorithms, memetic algorithms, heuristics, Transcription Factor Binding Sites

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

CFTR	*Cystic Fibrosis Transmembrane Conductance.*
CREB	*Cyclic-amp Response Element-Binding Protein.*
CRP	*Cyclic-amp Receptor Protein.*
ChIP	*Chromatin Immunoprecipitation.*
DBD	*DNA Binding Domain.*
DMEC	*Discovery Motifs by Evolutionary Algorithms.*
DMMA	*Discovery Motifs by Memetic Algorithms.*
DNA	*Deoxyribonucleic Acid.*
EA	*Evolutionary Algorithms.*
EP	*Evolutionary Programming.*
ES	*Evolutionary Strategies.*
GA	*Genetic Algorithms.*
GRASP	*Greedy Randomized Adaptative Search Procedure.*
HLH	*Helix Turn Helix.*
HNF1	*Hepatocyte Nuclear Factor-1.*
HTH	*Helix Turn Helix.*
MEF2	*Myocite Enhancer Factor-2.*
MA	*Memetic Algorithms.*
MFMD	*Memetic Framework for Motif Discovery.*
MyOD	*Myogenic Differentiation-1.*
MLSA	*Multiple Local Sequence Alignment.*
NFkB	*Nuclear Factor Kappa-ligth-chain-enhancer of activated B cells.*
PB	*Pares de Base.*

PM	*Palindromic Motifs.*
PFM	*Positon Frequency Matrix.*
PPM	*Position Probability Matrix.*
PSSM	*Position Specific Scoring Matrix.*
PWM	*Position Weight Matrix.*
RNA	*Ribonucleic Acid.*
RNAm	*RNA Mensageiro.*
RNAnc	*RNA Não-codificante.*
RNAr	*RNA Ribossomal.*
RNAt	*RNA Transportador.*
SA	*Simulated Annealing.*
SDM	*Space Dyad Motifs.*
SRF	*Serum Response Factor.*
SH3	*SRC Homology 3.*
ST	*Search Tree.*
TBP	*TATA-binding Protein.*
TF	*Transcription Factor.*
TFBS	*Transcription Factor Binding Sites.*
VNS	*Variable Neghborhood Search.*
WBA	*Word Based Algorithms.*
ZF	*Zinc Finger.*

SUMÁRIO

1 INTRODUÇÃO

A Biologia Molecular tem-se desenvolvido potencialmente desde a descoberta da estrutura química do DNA. Grande parte disso graças à possibilidade de serem aplicadas diferentes técnicas computacionais em seus domínios, contribuindo assim com o seu progresso. O modo como as moléculas se alinham ao longo de uma cadeia de DNA, permite-lhes serem tratados computacionalmente como sequências de símbolos de um alfabeto finito (8).

A ligação de proteínas específicas, chamadas de fatores de transcrição (TF, do inglês *Transcription Factors*), a locais distintos da sequência genômica é considerada um elemento fundamental no processo que envolve a regulação gênica, podendo conduzir a alterações na atividade transcricional para um particular gene alvo (9). Estas localizações (sítios de ligação), em geral, são curtas (< 30 bps) e possuem uma sequência comum de nucleotídeos, embora normalmente possam existir variações entre os sítios devido a mutações que ocorreram em virtude da pressão seletiva que o genoma sofreu ao longo do tempo (1).

Os sítios de ligação compartilham entre si, um padrão de sequência específico o suficiente para que fatores de transcrição não consigam se ligar a localizações aleatórias ao longo do genoma. Por outro lado, a especificidade não pode ser absoluta devido a necessidade de existirem diferentes afinidades de ligação entre as proteínas transcricionais e os seus respectivos alvos em diferentes genes (10).

O conjunto de sequências capazes de agir como sítios de ligação para um particular fator de transcrição são denominados *motivos*. Em muitas situações, as localizações dos *motivos* devem ser aprendidas sem conhecimento prévio. Neste caso o problema recebe a denominação de descoberta de *motivo de novo* ou do inglês *de novo motif discovery*.

Os *motivos* atuam como sítios de ligação para proteínas específicas e desempenham um papel essencial na ativação e repressão primária da expressão genica. Em geral, eles ficam localizados dentro das regiões promotoras dos genes, todavia, podem também estar contidos dentro de sequências exônicas, dentro de íntrons ou ainda na fita negativa do gene (11).

Ensaios biológicos como *DNA Footprinting* (12) e Eletroforese em Gel (13) são frequentemente os métodos mais confiáveis e precisos para identificar os Sítios de Ligação dos Fatores de Transcrição (TFBSs, do inglês *Transcription Factor Binding Sites*) entretanto,

a validação experimental é cara e dispendiosa, o que torna as abordagens computacionais uma alternativa atraente para a descoberta de *motivos de novo*.

Um conjunto de sequências de regiões cis-reguladoras pertencentes a genes co-expressos e ligadas pelo mesmo fator de transcrição podem ser coletadas com base nos seus padrões de expressão semelhantes. Desta forma, através da comparação destas sequências, é possível extrair subsequências semelhantes e descobrir os TFBSs subjacentes. Estes locais de ligação são denominados instâncias de *motivo* (14).

Em geral, as proteínas reguladoras apresentam-se como dímeros, pois desta forma conseguem se ligar mais fortemente ao DNA. Em particular, os TFs podem se unir tanto à fita positiva quanto à fita negativa do gene. Neste caso a sequência é dita palindrômica, isto é, a fita positiva do *motivo* é igual ao seu complemento reverso. Por exemplo, a enzima de restrição *EcoRI* do organismo *E. coli* é um homodímero que se liga a um sítio cuja sequência é definida por GAATTC (15).

Com o crescimento do número de genomas sequenciados, tornou-se essencial o nascimento de técnicas mais rápidas e baratas que conservassem um bom nível de confiabilidade na investigação dos dados gerados (5). Deste modo, as técnicas computacionais foram ganhando destaque na análise de sequências biológicas.

Existem diversas abordagens na literatura que tentam resolver este problema de forma eficiente (16) das quais podemos destacar os métodos probabilísticos e os métodos exatos (17).

Os métodos probabilísticos buscam maximizar a entropia relativa conhecida como *Kullback-Leibler divergence* (18), obtida a partir da construção de uma Matriz de Escore de Posição Específica (PSSM, do inglês *Position Specific Score Matrix*). Existem vários algoritmos dentro deste conjunto dos quais podemos citar: MEME (19), CONSENSUS (10) e Gibbs Motif Sampler (20). Estes algoritmos geralmente possuem um rápido tempo de convergência, porém, podem ficar "presos" em ótimos locais.

As abordagens exatas usualmente utilizam a sequência consenso para a representação dos *motivos*, empregando algum tipo de otimização matemática como modelo de busca. Em geral, estes algoritmos possuem um alto tempo de execução, em particular para longos tamanhos de *motivos* (21). Em contrapartida, eles conseguem fugir de ótimos locais devido a natureza exata de sua busca. Como exemplos podemos citar SPELLER (22) e WEEDER (23).

1.1 Justificativa

Os estudos a cerca dos *motivos* biológicos dividem-se basicamente em duas vertentes: a primeira leva em consideração a importância destes padrões dentro da biologia molecular, onde a sua compreensão é a chave para o entendimento de muitos eventos celulares (24). A segunda atende ao campo da biologia computacional, cuja importância se concentra na investigação de técnicas eficientes que maximizem a correta localização destes sinais dentro das sequências biológicas.

Existem diversas perguntas que podem ser respondidas através do estudo sistemáticos de *motivos*, das quais podemos destacar:

- identificação de sítios alvo para novas drogas;
- identificação dos sítio de ligação dos fatores de transcrição;
- inferência de redes de regulação gênica;
- identificação de sítios funcionais em proteínas;
- determinação de epítopos em antígenos.

O estudo sobre redes de *motivos* pode elucidar como ocorrem as interações entre os fatores de transcrição e os genes que eles regulam. Conhecendo esses mecanismos, os pesquisadores podem desenvolver formas de alterá-los, como por exemplo, através do uso de fármacos. A aquisição de resistência por parte das bactérias tem provocado a diminuição da atividade de importantes antibióticos, sendo um considerável objeto de pesquisa (25).

Algumas bactérias possuem genes que codificam proteínas capazes de clivar, quebrar ou promover alterações estruturais nas moléculas de alguns fármacos, tornando-os inativos. Um exemplo são as enzimas *b-lactamases*, produzidas por algumas bactérias resistentes, que inibem a ação dos antibióticos *b-lactâmicos*. Conseguir entender os mecanismos que agem por trás da produção desta enzima torna possível o desenvolvimento de drogas mais eficientes (26).

O estudo de *motivos* também ajuda a entender o funcionamento dos receptores de células T. Esses receptores possuem um sítio de ligação específico para o reconhecimento do antígeno. Encontrar a localização destas regiões contribui com o entendimento sobre alguns tipos de infecções e doenças autoimunes (27).

Além das citadas, existem diversas outras razões onde a localização e principalmente a confirmação de sítios putativos de *motivos* são importantes, tais como: definição estrutural

e funcional de proteínas, predição de estruturas secundárias de RNAs, análise filogenética, dentro outros.

Do ponto de vista computacional, as principais motivações que guiam o desenvolvimento de novas abordagens são: a fragilidade dos algoritmos computacionais existentes e o alto custo que as técnicas experimentais possuem. Embora hajam diversas abordagens computacionais na literatura, ainda não existe uma que seja capaz de resolver o problema integralmente. Isso se deve à alta complexidade do espaço de busca e à falta de um modelo matemático preciso que consiga representar os diversos tipos de *motivos* existentes na natureza.

1.2 Objetivos

O principal objetivo deste trabalho é investigar os aspectos que envolvem o problema de descoberta de *motivos* em sequências de DNA, estudar as principais técnicas computacionais responsáveis pela identificação dessas estruturas em regiões promotoras de genes co–expressados e desenvolver um preditor capaz de resolver este problema de forma eficiente utilizando algoritmos evolutivos e meméticos. Também é objetivo deste trabalho comparar a abordagem aqui desenvolvida com algoritmos do estado da arte bem como utilizar datasets sintéticos e reais para cumprir este propósito.

1.3 Definição do Problema

Embora existam diversas formulações deste problema, iremos começar com a definição canônica e mais geral da descoberta de *motivos* da seguinte maneira:

Seja $S = \{s_1, s_2, \ldots, s_n\}$ um conjunto de palavras de comprimento L provenientes de um alfabeto $\Sigma = \{A, C, G, T\}$ de nucleotídeos e um tamanho de motivo w conhecido. Sem perda de generalidade e para simplicidade analítica, podemos supor que o comprimento de todas as sequências sejam iguais. Em situações reais, temos que $0 < w \ll L$.

O propósito é encontrar o mais promissor padrão de subsequências $X^* = \{x_1, x_2, \ldots, x_n\}$ de tamanho w e suas respectivas posições iniciais em S. A escolha de um particular padrão é baseada na definição de uma ou mais funções de escore que medem a similaridade ou diferença estatística entre o padrão de *motivos* e suas respectivas ocorrências.

Existem diversos métodos para mensurar a qualidade de um *motivo*. As funções objetivo devem ser capazes de refletir de forma precisa a eficiência de uma modelagem. Uma função de avaliação inadequada, que não reflete a verdade biológica, não será capaz de fornecer uma boa solução mesmo que um algoritmo de otimização eficiente seja utilizado.

Neste trabalho foi utilizado o *Information Content Score* (11) e o *Complexity Score* (28) como funções objetivo. Elas serão melhor apresentadas na Capítulo 5.

Já foi provado que a definição canônica do problema de *motivos* é NP-Difícil mesmo com os pressupostos mais simplificados (29). O espaço de busca cresce exponencialmente de acordo com o tamanho da entrada e fica definido pela Equação 1.1:

$$E = (L - w + 1)^N \tag{1.1}$$

Onde E é o tamanho total do espaço de busca, L é o tamanho de cada sequência, w é o tamanho de um particular *motivo* e N é o número total de sequências pertencentes ao dataset.

A Equação 1.1 prevê apenas situações cujas sequências possuam apenas um *motivo*. Embora seja uma boa aproximação, em circunstâncias reais, é bastante provável que mais de um *motivo* faça parte de sua composição. Deste modo, a Equação 1.2 descreve de forma mais precisa o tamanho real do espaço de busca:

$$\gamma(n) = \frac{N!}{n!(N-n)!} \times E \tag{1.2}$$

Onde E é o resultado obtido pela aplicação da Equação 1.1, n é o número de *motivos* no alinhamento, L é o tamanho de cada sequência, w é o tamanho de um particular *motivo* e N é o número total de sequências pertencentes ao dataset.

1.4 Organização do trabalho

Este trabalho está organizado como segue:

O Capítulo 2 apresenta os principais conceitos da biologia molecular. O Capítulo 3 revisa sobre os vários tipos de *motivos* e as principais abordagens computacionais utilizadas na literatura para descoberta dessas estruturas. Este capítulo também discute quais são os principais desafios e dificuldades encontrados no desenvolvimento de abordagens eficientes. O Capítulo 4 descreve os principais conceitos da computação evolutiva e algoritmos meméticos bem como apresenta a descrição das heurísticas utilizadas neste trabalho. O Capítulo 5 descreve a metodologia proposta, a forma como os algoritmos foram construídos e as técnicas utilizadas no desenvolvimento da abordagem. O Capítulo 6 apresenta os resultados, tabelas e gráficos obtidos a partir dos testes realizados no conjuntos de dados. Por fim o Capítulo 7 faz a conclusão e a define as possíveis direções futuras da pesquisa.

2 CONCEITOS BIOLÓGICOS

As células possuem características estruturais muito similares, cujo conteúdo intracelular é separado do meio extracelular por uma membrana plasmática. Na sua parte interna, as organelas (no caso de organismos eucariontes) e diversas outras partículas ficam dispersas em uma complexa solução chamada de citosol (30).

Embora as células apresentem muitas similaridades, os organismos detêm substanciais diferenças a nível celular, podendo ser classificados como *Bacteria*, *Archea* e *Eukarya*. Organismos procariotos são sempre unicelulares e podem ocorrer na forma de colônias (31). Segue definição segundo ZAHA et al. (30).

> De maneira geral, as células são unidades de vida compartimentalizadas, formadas por um complexo agregado de moléculas, organizadas conforme suas funções e delimitadas por uma bicamada molecular, as membranas celulares.

Em algum momento de sua existência, todas as células possuíram um nucleoide ou núcleo, onde o genoma é replicado e armazenado com suas proteínas associadas. Em bactérias e em arqueias, o nucleoide não é separado do citoplasma por uma membrana. Estes microrganismos sem membrana nuclear, antes classificados como procariontes, são agora reconhecidos como pertencentes a dois grupos: *Bacteria* e *Archea*. Por outro lado, nos eucariotos, o DNA é confinado dentro de um envelope nuclear; este grupo compõe o grande domínio dos *Eukarya* (31).

2.1 Biomoléculas

Embora a água seja o elemento mais abundante das células, existem outros componentes químicos importantes que também fazem parte de sua constituição, como as biomoléculas. As biomoléculas são compostos de carbono com uma grande variedade de grupos funcionais e compreendem as moléculas pequenas e as macromoléculas (31).

Existe uma coleção de aproximadamente mil moléculas pequenas dissolvidas no citosol, que incluem metabólitos centrais das principais rotas metabólicas, como aminoácidos comuns, nucleotídeos, açucares e seus derivados fosforilados e ácidos mono, di e tricarboxílicos. O grupo das macromoléculas, também chamadas de polímeros biológicos, é constituído por proteínas, ácidos nucleicos e polissacarídeos (31).

É importante notar que apesar dos lipídeos não serem polímeros, eles estão presentes em grande quantidade nas células. Os lipídeos são constituídos por ácidos graxos e possuem diversas funções, além de participarem da composição da membrana celular (30). Podemos destacar quatro tipos básicos de moléculas que contribuem na síntese dos polímeros: os nucleotídeos, os aminoácidos, os açucares e os ácidos graxos (Figura 1).

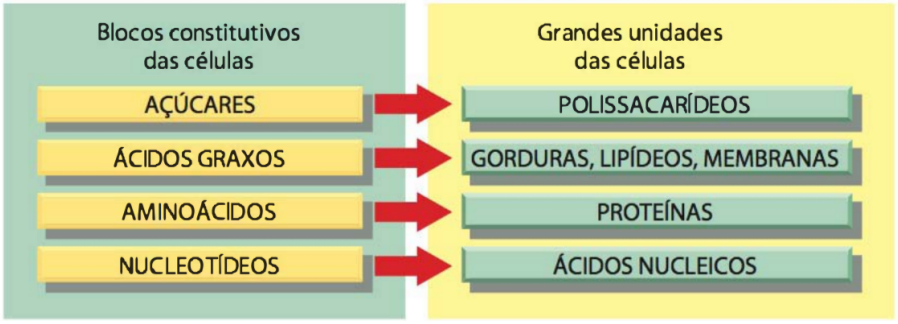


Figura 1 – As quatro moléculas menores e seus respectivos polímeros. À direita temos os blocos monoméricos ou subunidades formadoras das macromoléculas que estão localizadas à esquerda na figura. É importante lembrar que os lipídeos, apesar de serem polímeros, não são considerados macromoléculas. Fonte: (1).

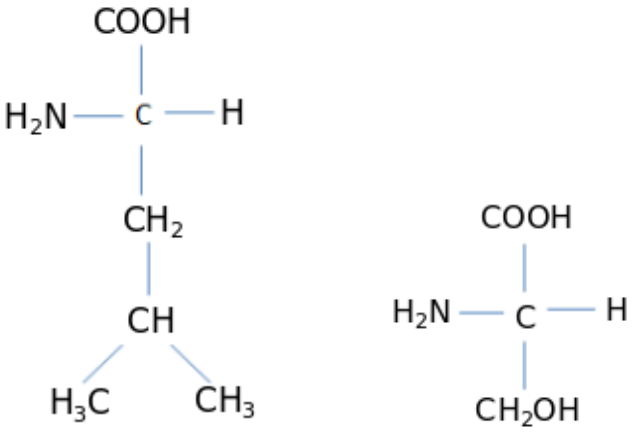
Basicamente, os polissacarídeos são polímeros de açúcares mais simples e podem apresentar três funções principais como: depósito de combustível de alto conteúdo energético, componentes estruturais rígidos da parede celular (em plantas e bactérias) e elementos no reconhecimento celular extracelular que se ligam a proteínas de outras células. Os lipídeos são derivados de hidrocarbonetos, são insolúveis em água e atuam como componentes estruturais da membrana celular, como depósitos energia, como pigmentos, sinalizações intracelulares e hormônios (31).

As proteínas são polímeros longos de aminoácidos e podem ter função catalítica (enzimas), estrutural, como receptoras de sinais ou transportadoras de substâncias para dentro e fora da célula, dentre outras. Os ácidos nucleicos, DNA e RNA, são polímeros de nucleotídeos que armazenam e transmitem a informação genética, e algumas moléculas de RNA apresentam também função estrutural e catalítica. Devido às suas características ricas em informação, as proteínas e os ácidos nucleicos, em particular o DNA, também são referenciados como macromoléculas informacionais (31), são o foco deste estudo e serão especificamente abordadas nas seções seguintes.

2.1.1 Proteínas

As proteínas são macromoléculas biológicas sintetizadas pela célula a partir moléculas menores chamadas de aminoácidos. Elas podem adotar uma enorme variedade de arranjos tridimensionais, o que lhes compete um aspecto multifuncional (5).

As proteínas são o produto direto da tradução do RNA. Existem 20 aminoácidos que podem ser codificados diretamente pelo genoma e outros que podem ser sintetizados a partir destes, como é o caso da selenocisteína. Os aminoácidos possuem uma mesma estrutura geral (Figura 2), contendo um grupo amina, um grupo carboxila e uma cadeia lateral de tamanho variável, chamada de grupo R ou radical. Eles diferem uns dos outros pela estrutura e propriedades do grupo R.



(a) (b)

Figura 2 – Estrutura química simplificada dos aminoácidos (a) Leucina e (b) Serina. Na parte superior está presente o grupo carboxila CO_2H, à esquerda o grupo amina NH_2 e na parte inferior o radical.

Os aminoácidos podem se unir através de ligações covalentes, chamadas de ligações peptídicas, formando assim as cadeias polipeptídicas (Figura 3). Nestas cadeias lineares, o carbono do grupo carboxila liga-se ao nitrogênio do grupo amina formando o que chamamos de proteínas.

As proteínas estão ligadas as mais variadas funções celulares, desde o transporte de nutrientes e metabólitos à catalise de reações biológicas. Sua composição é relativamente simples em relação à sua complexidade e número de funções. Em organismos superiores, elas representam cerca de 50% do peso seco dos tecidos (1).

Seu papel é fundamental e praticamente todos os processos biológicos dependem da presença ou da atividade dessa biomolécula. Alguns exemplos de funções desempenhadas por elas são: enzimas, hormônios, proteínas transportadoras, anticorpos, receptores e estruturas. Elas são classificadas de acordo com sua estrutura e formas de enovelamento. Esses diferentes níveis de apresentação são nomeados como estruturas primárias, secundária, terciária e quaternária.

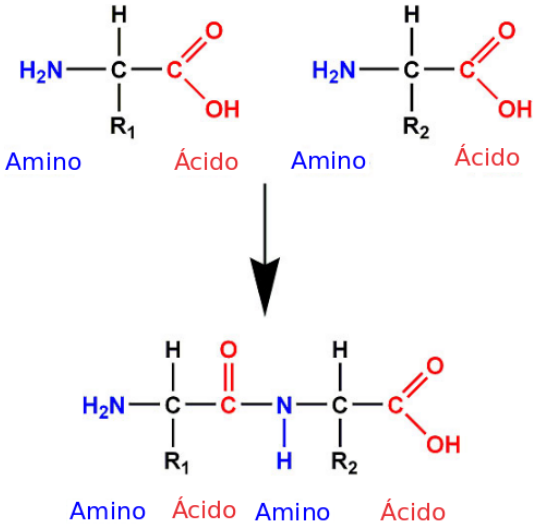


Figura 3 – Uma cadeia polipeptídica é formada pela ligação entre vários aminoácidos. A parte superior da figura mostra dois aminoácidos justapostos. Na parte inferior, é possível observar o carbono do grupo carboxila do primeiro aminoácido se ligando ao nitrogênio do grupo amina do segundo através de uma ligação covalente peptídica.

2.1.2 Ácidos nucleicos

Os ácidos nucleicos são macromoléculas sintetizadas a partir de unidades monoméricas menores chamadas de nucleotídeos (5). Cada nucleotídeo é formado por três unidades básicas: uma base nitrogenada(púrinica ou pirimídica), um grupo fosfato e um açúcar (pentose). No DNA, a pentose recebe o nome de 2-desoxi-β-D-ribose ou como é mais conhecida desoxirribose e possui a seguinte fórmula molecular $C_5H_{10}O_4$. No RNA, a pentose é ligeiramente diferente, possuindo um oxigênio a mais em sua fórmula $C_5H_{10}O_5$. Por essa razão, a pentose do RNA recebe o nome de β-D-ribose ou ribose. São os ácidos nucleicos que armazenam as informações de que as células precisam para codificar as proteínas, por exemplo (30).

Os ácidos nucleicos são unidos através de ligações covalentes entre a pentose e o grupo fosfato. A essa ligação é dado o nome de fosfodiéster. Dada a capacidade dos carboidratos de deformar seu anel e a flexibilidade da ligação fosfodiéster, o esqueleto dos ácidos nucleicos tende a ser bastante flexível e resistente. Entre as bases nitrogenadas existem ligações de hidrogênio que podem ser duplas ou triplas. As bases nitrogenadas são de cinco tipos, cada qual com uma estrutura química diferente: adenina (A), guanina (G), citosina (C), uracila (U) e timina (T). As bases adenina, citosina e guanina são encontradas tanto no DNA quanto no RNA. A timina somente no DNA e a uracila somente no RNA. Existem ainda outras bases menos comuns e com características químicas distintas que são encontradas somente no RNA, tais como a hipoxantina e a inosina (31).

Na Figura 4 é possível observar as bases nitrogenadas mais comuns e suas respectivas

estruturas químicas. No DNA, a adenina se liga a timina e a guanina à citosina. No RNA temos a adenina formando ligações com a uracila e a guanina com à citosina. Em ambos os casos temos bases purínicas se ligando as bases pirimidínicas, como mostra a Figura 5.

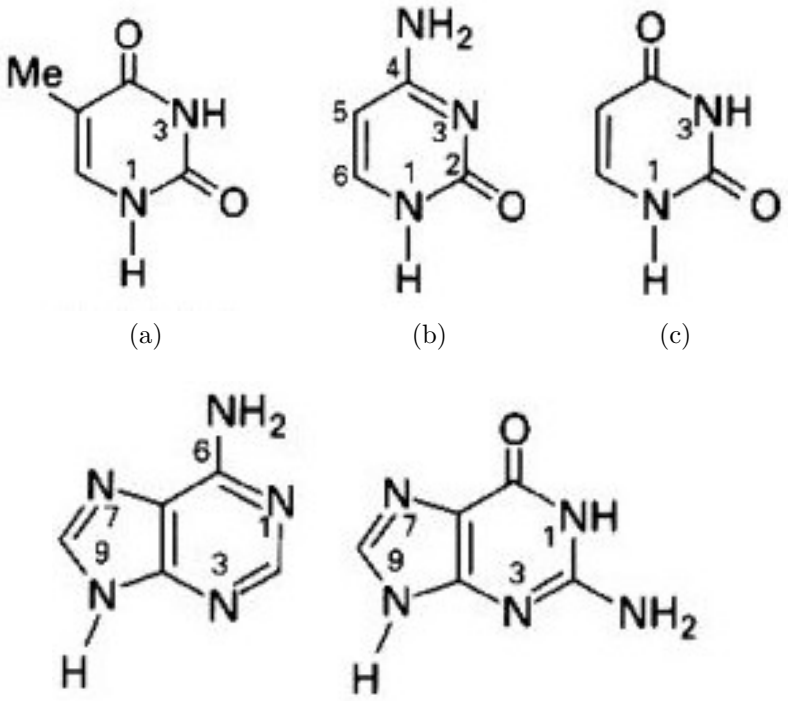


Figura 4 – Na parte superior da figura temos as pirimidinas (a) timina, (b) citosina, (c) uracila. Abaixo temos as purinas (d) adenina e (e) guanina.

Uma característica muito importante dos ácidos nucleicos é a orientação de suas ligações. Em uma de suas extremidades, existe um grupamento fosfato ligado ao carbono 5 (5') da pentose. Na outra extremidade, existe uma hidroxila ligada ao carbono 3 (3') do açúcar. Desta forma, convencionou-se que a orientação das ligações dos ácidos nucleicos acontecem no sentido 5' $\longrightarrow$ 3' (lê-se 5 linha 3 linha), também chamado de "sentido da vida", pois, a polimerização dos monômeros acontecem neste sentido (1).

2.1.2.1 DNA

A molécula de DNA é constituída por duas longas cadeias ou fitas polinucleotídicas, com giro para à direita, compostas por quatro tipos de subunidades monoméricas (1). Com exceção dos vírus, todos os organismos armazenam suas informações genéticas no DNA (32). As fitas se mantêm unidas em dupla hélice por pontes de hidrogênio entre as bases nitrogenadas. O DNA possui as bases adenina (A), guanina (G), citosina (C) e timina (T), onde:

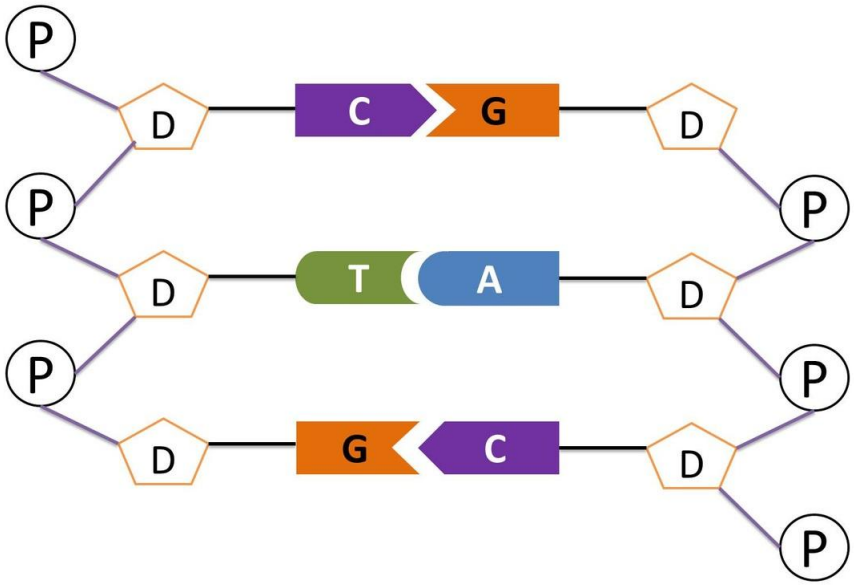


Figura 5 – Esquema simplificado da estrutura primária de um ácido nucleico (DNA). Nas laterais da figura é possível observar as ligações entre o grupamento fosfato (P) e o açúcar (D), formando a chamada ligação fosfodiéster. No interior da figura, temos as bases purínicas (A e G) formando ligações com as bases pirimidínicas (C e T). Fonte: (2)

- A e G têm estrutura de dois anéis, característica de um tipo de substância chamada purina;
- C e T têm a estrutura com somente um anel, característica relacionada às pirimidinas.

As bases nitrogenadas e o grupamento fosfato estão ligadas ao primeiro e ao quinto carbono do desoxirribose respectivamente. Um desoxirribonucleotídeo é formado por um nucleotídeo, uma desoxirribose e um ou mais grupos fosfato. Eles são ligados entre si por ligações covalentes chamadas de ligações fosfodiéster que são estabelecidas entre o grupamento fosfato (5'P) e o grupo hidroxila (3'OH) (30).

A polaridade química da fita de DNA está relacionada com a maneira que seus nucleotídeos estão ligados. O DNA possui uma estrutura tridimensional em forma de dupla hélice. Ela se constitui em decorrência das características estruturais e químicas que sua cadeia polinucleotídica possui. As bases nitrogenadas ficam voltadas para o interior da dupla hélice e o esqueleto de açúcar-fosfato encontra-se voltado para o exterior. As bases purínicas, que possuem dois anéis em sua composição, sempre se pareiam com bases pirimidínicas, que possuem apenas um anel (31).

Esse tipo de arranjo maximiza a energia da molécula e permite que cada par de bases possua uma largura similar, mantendo a estrutura de açúcar-fosfato equidistante ao

longo da molécula de DNA (1). Em outras palavras, a união entre adenina (A) e timina (T), e guanina (G) e citosina (C) permite que ligações eficientes de hidrogênio sejam formadas. Com isso, mesmo que haja aproximação entre os átomos, a dupla hélice se mantém estável. Isso somado a algumas interações hidrofóbicas, permite a molécula de DNA manter a sua conformação tridimensional (30).

Existe uma restrição para que os elementos de cada par de bases consigam se "encaixar" na dupla-hélice. Isso só ocorre se a polaridade de uma fita estiver em orientação oposta à da outra fita. Desta forma, para que haja pareamento, cada fita deverá estar em posição paralela à outra, porém, de forma invertida. Por isso, diz-se que a dupla fita do DNA é antiparalela (31).

2.1.2.2 RNA

O RNA é um polímero que, ao contrário do DNA, é formado por apenas uma cadeia polinucleotídica. Sua estrutura primária (Figura 6) é muito parecida com a do DNA, exceto por duas mudanças: a primeira é o açúcar, que no DNA é uma desoxirribose e no RNA é uma ribose. A segunda consiste na troca da base nucleotídica timina (T), presente no DNA para uracila (U) no RNA. Embora possuam organizações parecidas, o DNA possui estrutura molecular mais estável em relação RNA (30).

O dogma central da biologia molecular descreve como a mensagem armazenada no DNA é transcrita na forma de RNA, para que, posteriormente, seja traduzida em proteína. Existem basicamente três classes principais de ácidos ribonucleicos:

- RNA mensageiro (mRNA): contém a informação genética que irá ser codificada em aminoácidos. Em outras palavras, possui as informações necessárias para síntese de proteínas. Representa de 1 a 5% do total de RNA de uma célula;
- RNA transportador (tRNA): possui a tarefa de identificar e transportar os aminoácidos presentes nas células até os ribossomos para a síntese de proteínas; Representa de 10 a 15% do total de RNA de uma célula;
- RNA ribossômico (rRNA): provê todas as condições necessárias para a síntese das proteínas. Os ribossomos são sintetizados em uma região do núcleo chamada de nucléolo. Logo após serem sintetizados, se unem a proteínas específicas e se dirigem ao citoplasma passando através dos poros da membrana nuclear. Representam por volta de 80% do total de RNA de uma célula;
- RNA não-codificante (ncRNA): são todos os RNAs que não são traduzidos em proteínas. Os genes a partir dos quais o ncRNA é transcrito são chamados de genes

de RNA ou genes de RNA não-codificado. Representam todos os tipos de RNAs com exceção do mRNA.

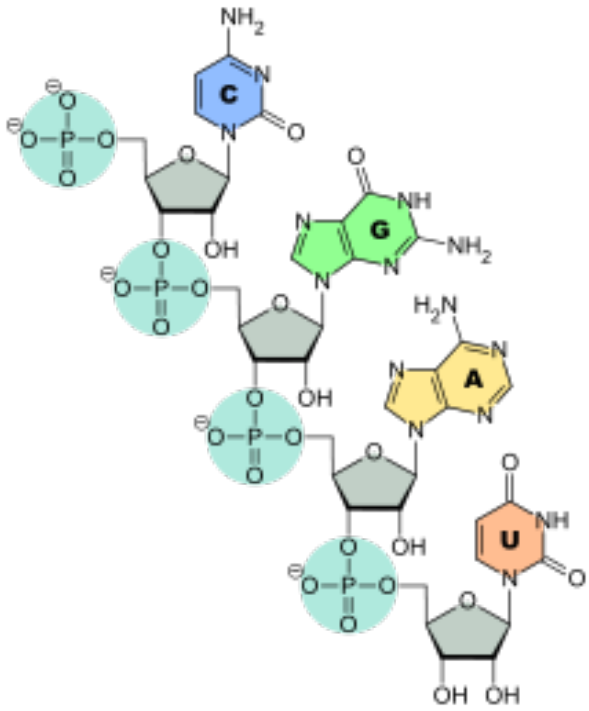


Figura 6 – Esquema da estrutura primária de uma molécula de RNA. Assim como no DNA, o esqueleto da molécula de RNA é constituída de ligações fosfodiéster. A diferença está no número de cadeias que o RNA possui, apenas uma. Uma outra diferença entre eles está no tipo de bases nitrogenadas que constitui cada molécula. No RNA, a timina (T) da lugar a uracila (U). Fonte: (3).

Embora a estrutura primária do RNA seja relativamente simples, sua estrutura secundária pode ser muito complexa. O RNA possui algumas regiões complementares que podem enovelar-se e assumir vários tipos de conformações. As bases adenina (A) e uracila (U) assim como guanina (G) e citosina (C) formam pontes hidrogênio entre si e como resultado, o RNA constrói inúmeras conformações tridimensionais na forma de alças e grampos. Assim como nas proteínas, é a conformação tridimensional que confere função ao RNA.

A molécula de RNA é o produto da transcrição da informação genética. As células conseguem controlar onde e quando a transcrição deverá acontecer. De um modo geral, ela se inicia em uma região chamada de região promotora e termina em uma região chamada de terminador.

3 *MOTIVOS* BIOLÓGICOS

Os ácidos nucleicos e as proteínas possuem determinadas regiões que, ao longo da evolução, experimentaram de maneira diferente a pressão seletiva (33). Presume-se que essas regiões tipicamente possuam algum significado biológico para a célula e a sua compreensão pode ajudar a esclarecer o funcionamento organismo como um todo (34).

São vários os incentivos que guiam as pesquisas relacionadas a *motivos*. Seu estudo torna possível identificar com maior precisão os sítios de ligação dos fatores de transcrição (TFBS, do inglês *Transcription Factor Binding Sites*) e nucleases, sítios ativadores e silenciadores, sítios de ligação do ribossomo, sítios de termino da transcrição, sítios de *splicing* alternativo e de poliadenilação, sítios ativos das enzimas, sítios de ligação de nucleotídeos, domínios proteicos, dentre outros.

3.1 *Motivos* em proteínas

Com o crescimento exponencial do número de genomas sequenciados, os métodos de classificação baseados em famílias de proteínas se tornaram um relevante problema a ser solucionado (35). Em proteínas, os *motivos* são pequenas regiões conservadas dentro de longas cadeias de aminoácidos. A estrutura tridimensional de uma proteína depende das interações bioquímicas realizadas entre os aminoácidos de sua sequência e também das relações que eles mantém com o ambiente que os envolvem (36). Essa conformação tridimensional permite as proteínas interagir com outros compostos e determinam sua função dentro do organismo. A forma exata do enovelamento de uma proteína é definida por:

- interações não-covalentes – ligações de hidrogênio, atrações eletrostáticas e atrações de *Van Der Waals* – entre os aminoácidos de sua cadeia;
- interações dos aminoácidos apolares (hidrofóbicos) com o meio aquoso, fazendo com que eles fiquem agrupados no interior da molécula.

É importante notar a diferença entre *motivos* e domínios proteicos. O termo *motivo* é definido como um padrão de enovelamento identificável, envolvendo dois ou mais elementos da estrutura secundária e a conexão (ou conexões) entre eles. Um domínio proteico é uma parte da cadeia polipeptídica que é independentemente estável e pode se movimentar como uma entidade isolada em relação ao resto da proteína (31).

No entanto, as sequências de um mesmo domínio podem ser pouco similares e ainda assim desempenhar a mesma função. Em outras palavras, existe uma combinação enorme de sequências de aminoácidos que permitem formar a mesma estrutura tridimensional de um domínio. Isso ocorre pela ação das forças de ligação citadas acima e também porque os aminoádos podem ter propriedades físico-quimicas semelhantes. Desta forma, a busca por um domínio proteico somente pela análise de suas sequências pode ser ineficiente (5).

Como exemplo, podemos citar o domínio SRC homologo 3 ou simplesmente SH3 (Figura 7). O SH3 é um pequeno domínio proteico com cerca de 60 resíduos cuja atividade envolve interações proteína-proteína. Domínios iguais de famílias diferentes podem possuir sequências altamente divergentes. Isso porque, ao longo do processo evolutivo, a estrutura tridimensional das proteínas tende a se preservar, pois do contrário, perderiam suas funções. No entanto, domínios proteicos pertencentes a uma mesma família tipicamente possuem estrutura primária mais conservada.

Nas proteínas, também podemos encontrar os *motivos* estruturais. Eles estão associados as conformações locais dentro de uma cadeia polipeptídica provenientes das interações entre as folhas-β, hélices-α e *turns*. Uma proteína é formada pela união de várias estruturas secundárias interligadas. Essas estruturas formam ligações não-covalentes entre si dando origem aos chamados *folds* (Figura 8). As estruturas secundárias são tipicamente classificadas em:

- α: possui apenas hélices-α;
- β: possui apenas folhas-β;
- α/β: folha-β e hélices-α ocorrendo de forma alternada;
- $\alpha + \beta$: folha-β e hélices-α ocorrendo em regiões separadas.

Os *motivos* estruturais são classificados de acordo com a sua função e estrutura tridimensional. Em geral, temos as superfamílias que compreendem as proteínas com características estruturais semelhantes, porém com funções diferentes. As superfamílias possuem agrupamentos de famílias de proteínas que possuem estrutura e funções muito semelhantes.

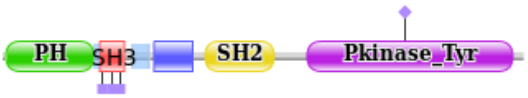
As combinações regulares da estrutura secundária também podem formar os sítios de ligação de nucleotídeos (DBD, do inglês *DNA-binding domain*). As proteínas que possuem esses sítios são chamadas de proteínas de regulação gênica ou fatores de transcrição (TF, do inglês *Transcription Factor*). Elas podem se ligar a determinadas regiões de um gene e permitir que ele seja transcrito (1). Um exemplo é o fator *sigma* 70 (σ 70 *Transcription*

(a)

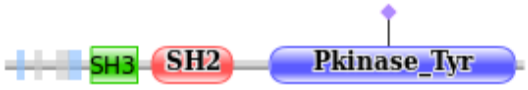
```
>BLK_MOUSE/58-104
VALFDYAAVNDRDLQVLKGEKLQVLRSTG-DWWLARSLVTGREGYVPS
>TEC_HUMAN/185-231
VAMYDFQAAEGHDLRLERGQEYLILEKNDVHWWRAR-DKYGNEGYIPS
>TXK_HUMAN/88-134
KALYDFLPREPCNLALRRAEEYLILEKYNPHWWKAR-DRLGNEGLIPS
```

(b)

(c)



(d)



(e)

Figura 7 – (a) Estrutura tridimensional do domínio SH3. Em geral, a estrutura tridimensional de um domínio proteico é conservada, mesmo possuindo algumas divergências em suas estruturas primárias. (b) Alinhamento múltiplo dos domínios SH3 das proteínas (c) *Tyrosine-protein kinase Blk*, (d) *Tyrosine-protein kinase Tec* e (e) *Tyrosine-protein kinase TXK*. A primeira pertence ao organismo *Mus musculos* (camundongo) e as duas últimas pertencem ao *Homo sapiens*. As três proteínas fazem parte da mesma família (SH3_1) e portanto suas sequências possuem trechos conservados. Fonte: (4).

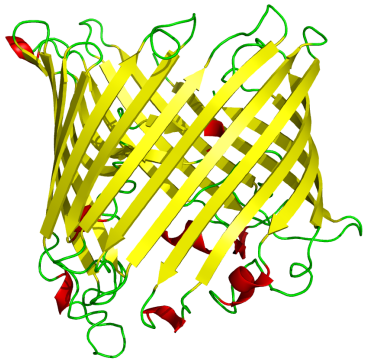


Figura 8 – Conformação do *motivo* estrutural barril-β. Seu enovelamento tridimensional lembra o formato de um barril. Isso permite a composição de uma cavidade central que pode servir como carregador de substâncias ou poros. Fonte: (5).

Factor), encontrado principalmente em bactérias. Esse fator se liga a uma subunidade da holoenzima RNA polimerase direcionando-a a uma classe específica de regiões promotoras. Essa ligação permite a RNA polimerase transcrever os principais genes basais de uma bactéria.

As proteínas de regulação gênica, tal como os fatores *sigma*, possuem *motivos* estruturais em sua superfície que podem ler pequenos trechos de sequências de DNA, formando vários tipos de ligações com os nucleotídeos (ligações de hidrogênio, ligações iônicas e interações hidrofóbicas). Estudos de cristalografia por raios X e de espectroscopia por NMR revelaram que as proteínas reguladoras possuem em suas superfícies alguns *motivos* estruturais compostos por folhas-β e hélices-α que se ligam ao sulco maior do DNA (1). Os principais tipos são:

- Hélice-volta-Hélice (HTH, do inglês *Helix-turn-Helix*);
- Dedo de zinco (ZF, do inglês, *Zinc Finger*);
- Zíper de leucina;
- Hélice-alça-Hélice (HLH, do inglês *Helix-loop-Helix*);

3.2 *Motivos* em ácidos nucleicos

Os *motivos* presentes em ácidos nucleicos são, em geral, pequenos padrões recorrentes de sequências nucleotídicas, com um suposto significado biológico (15).

Assim como nas proteínas, é preciso diferenciar os *motivos* estruturais dos *motivos* sequenciais. Nem sempre é possível encontrar alguma relação entre eles. Embora possam ser encontrados trechos conservados de sequências em *motivos* estruturais de uma mesma

família, a relação entre estrutura primária, secundária e terciária muitas vezes é frágil. Isso também acontece com as moléculas de RNA.

Depois da descoberta do funcionamento do *operon lac* e da constatação de que ele é regulado por um fator de proteína, a biologia molecular concentrou seus esforços no entendimento dos sítios de ligação dos fatores de transcrição (8). Isso possibilitou um estudo mais detalhado das redes de regulação gênica, na qual uma única proteína pode regular positiva ou negativamente a transcrição de vários mRNAs.

Estudos mais detalhados dos transcritos gênicos expandiram o entendimento sobre os RNAs não codificantes (ncRNA) (37). O que antes era chamado de "RNA-lixo", passou a ter um importante papel nos sistemas regulatórios celulares. A análise comparativa entre sequências é uma das formas mais confiáveis na predição da estrutura secundária do RNA e o estudo de múltiplas sequências correlatas pode ajudar a inferir um consenso sobre elas (37).

3.2.1 *Motivos* em DNA

Para que um gene possa ser transcrito, proteínas específicas, chamadas de fatores de transcrição (FT), precisam reconhecer e se ligar a determinadas regiões do gene chamadas de regiões regulatórias. Devido ao movimento das moléculas dentro da célula, os FTs podem unir-se acidentalmente à outras regiões do DNA, porém, eles possuem uma afinidade substancialmente maior (de 100 mil à 10 milhões vezes) de se ligarem as regiões regulatórias.

Um gene possui diversas regiões regulatórias, no entanto, a região promotora é a principal delas. Ela fica localizada na porção 5' UTR de um gene e é nela que está localizado um importante *motivo* chamado de sítio de ligação dos fatores de transcrição (TFBS, do inglês *Transcription Factor Binding Sites*) (15).

Conseguir identificar corretamente os TFBS, é um passo fundamental para o entendimento dos circuitos regulatórios que controlam a expressão genica (38). Um exemplo é o gene *operon lac* de *Escherichia coli*. Ele foi um dos primeiros sistemas complexos de regulação genica desvendados. Sua função é ajudar a transportar e metabolizar a lactose na ausência da glicose (8).

Existem dois tipos especiais de TFBS que podem ser encontrados nas regiões reguladoras: Os *palindromic motifs* (PM) e os *spaced dyad motifs* (SDM) (16). Os PMs são subsequências que possuem exatamente os mesmos nucleotídeos do seu complemento reverso. Por exemplo, o fator de transcrição EcoRI é um homodímero que se liga a um sítio no DNA cuja sequência é definida por GAATTC (15). Os SDMs são definidos por duas pequenas cadeias de nucleotídeos conservadas, que estão separadas por uma lacuna

(*gap*). Isso indica que um determinado fator transcrição se liga a um sítio SDM como um dímero, posicionando seus dois sítios de ligação na cadeia de DNA. No entanto, o espaço existente entre as subunidades de ligação do fator de transcrição conferem ao *motivo* seu formato.

Alguns genes possuem um sistema de regulação relativamente simples, porém, outros podem ser extremamente complexos. Os genes dos organismos eucariotos possuem, além da região promotora, outras regiões de regulação chamadas de sítios ativadores e repressores. Esses sítios são *motivos* onde proteínas específicas se unem para controlar a expressão genica. A regulação pode ser positiva ou negativa, dependendo do gene e da condição fisiológica da célula.

As enzimas de restrição são proteínas que se ligam à sítios específicos do DNA e possuem função de clivagem (15). Algumas bactérias podem ser infectadas por vírus bacteriófagos. Em geral, este tipo de vírus possui um mecanismo capaz de "injetar" seu DNA dentro da célula bacteriana. Uma vez dentro da célula, o DNA viral consegue unir-se ao DNA da bactéria e replicar-se.

Porém, as bactérias possuem um primitivo, porém, eficiente sistema imunológico capaz de reconhecer quais regiões estão infectadas (15). As proteínas de restrição reconhecem sítios altamente específicos do DNA bacteriano onde se ligam e clivam o DNA invasor. Estes sítios são representados por *motivos* altamente conservados, pois, se mutações ocorrerem nestes sítios, as enzimas de restrição podem clivar erroneamente seu próprio DNA.

As redes de regulação de transcritos são um dos vários fatores responsáveis pelo controle da expressão gênica organismos (25). Elas descrevem as interações que ocorrem entre os fatores de transcrição e os genes que eles regulam (25). Essas redes de regulação possuem pequenos conjuntos de padrões que podem ser descritos como *motivos*. Em *E. coli*, por exemplo, *motivos* interligados através de redes complexas foram detectados com muito mais frequência do que seria esperado em redes aleatórias. A Figura 9 ilustra detalhes dos sítios de ligação da proteína reguladora *AP-2-apha*.

Nesta Figura, é importante notar a construção da Logo da Sequência. Essa estrutura é gerada a partir do alinhamento múltiplo realizado entre os *motivos* de um determinado conjunto de dados. Em datasets com poucas amostras, o conteúdo da informação tende a ser superestimado. Por essa razão, uma correção no cálculo de entropia deve ser aplicado. Na geração da Logo, a quantidade da informação que cada nucleotídeo carrega é calculada utilizando a entropia relativa de Kullback-Leibler também chamada de distância de Kullback-Leibler (18). Essa medida leva em consideração tanto a frequência das bases presentes nos *motivos* quanto a frequência das bases que constituem as sequências negativas,

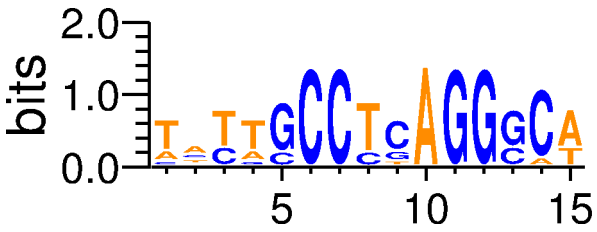
também conhecidas como sequências de fundo.

```
hg19_chr9:20618734-20618748(+)          TGTTCCCTCAGGGCA
hg19_chr12:51422769-51422783(+)         TTTTGCCTCAGGGAA
hg19_chr6:52149037-52149051(+)          TATAGCCCTAGGGCA
hg19_chr12:122583961-122583975(-)       AATGCCCCCAGGGCA
hg19_chr12:57577005-57577019(-)         CCTAGCCTGAGGCCA
hg19_chr6:53302614-53302628(+)          TTTTGCCTGAGGCCT
hg19_chr9:86577292-86577306(+)          AACTGCCTCAGGGCA
hg19_chr9:133956248-133956262(-)        TACTGCCTCAGGCCT
hg19_chr6:46843978-46843992(-)          TCCTGCCTCAGGGCT
                                             **  *** .:
```

(a)

```
A 2 4 0 2 0 0 0 0 0 9 0 0 0 0 6
C 1 2 3 0 2 9 7 2 6 0 0 0 3 8 0
G 0 1 0 1 7 0 0 0 2 0 9 9 6 0 0
T 6 2 6 6 0 0 0 7 1 0 0 0 0 0 3
```

(b)



(c)

TATTGCCTCAGGGCA

(d)

Figura 9 – Sítios de ligação do fator de transcrição *AP-2-apha*. Ele está envolvido na ativação de genes responsáveis pelo desenvolvimento dos olhos, face e tubo neural. Pode também agir suprimindo a expressão de vários genes, como MCAM/MUC18, C/EBP alpha and MYC. (a) Alinhamento de nove sítios de ligação conhecidos localizados em três cromossomos de *H. sapiens*. (*) Resíduos totalmente conservados. (:) Conservação entre os grupos com alta similaridade. (.) Conservação entre os grupos com baixa similaridade. (b) Matriz de frequência absoluta. Cada coluna desta matriz representa a quantidade de nucleotídeos encontrados em cada coluna do alinhamento múltiplo. (c) Logo da sequência. Representa o conteúdo de informação das frequências de cada par de base em suas respectivas posições (6). (d) Sequência consenso.

3.2.2 *Motivos* em RNA

As moléculas de RNA podem ser descritas como estruturas modulares constituídas por subunidades conservadas (39). Da mesma forma que as proteínas, os RNAs não codificantes, podem enovelar-se em diversas conformações tridimensionais, que conferem à

molécula diferentes funções na célula. Uma forma estrutural bastante comum e importante que o RNA pode assumir é a *A-form double helix* (40).

Existem diversas estruturas de RNA não codificantes na célula. Detre elas as principais são: RNAs transportadores (tRNA), RNAs Ribossomal (rRNA), micro RNAs (miRNAs), *small infering* RNAs (siRNAs), *small nuclear* RNAs. Em geral, praticamente todos os RNAs não codificantes (ncRNAs), salvo os tRNAs e rRNAs, possuem algum tipo de papel regulatório.

Embora o RNA seja uma molécula de fita simples, ele pode emparelhar suas bases e formar estruturas no formato de hélices ou *loops*. Esse processo pode ocorrer com ou sem *pseudoknot* [1]. Análises de variações de sequências realizadas em RNAs de uma mesma família podem ser utilizados para a inferência da estrutura secundária e terciária do RNA (39). Esta abordagem quando comparada com análises cristalográficas demonstram bons resultados (41).

A estrutura tridimensional de um RNA é bastante complexa de se determinar e experimentos biofísicos são necessários para uma descrição completa. Esta dificuldade está associada com a formação de ligações de longo alcance entre a bases do RNA, como: *pseudoknot*, zíperes de ribose, *kissing loops*, interações entre receptores *tetraloops-tetraloops*, hélices coaxiais, e outras ainda não caracterizadas. Além disso, o RNA pode sofrer interações de outras moléculas (co-fatores), tais como, metais e íons.

Os *motivos* estruturais, normalmente, possuem um *hairpin loop* ou *internal loop*. Eles são caracterizados ou identificados por sequências de nucleotídeos compatíveis com sua estrutura e função e podem ser de vários tipos: *tetraloops*, *sarcinricin loop*, *kink-turn* e *T-loop*.

Em geral, esses *motivos* são formados por pequenas estruturas tridimensionais conservadas que são chamadas de elementos estruturais. Eles são observados em vários tipos diferentes de *motivos* e tipicamente possuem sequências características. Estas pequenas sequências podem ser vistas como “assinaturas”, que são utilizadas para determinar a estrutura secundárias de um *motivo*.

Além dos *motivos* estruturais, outros tipos também são inferidos no RNA, como os sítios de ligação de proteínas e outras moléculas (39). Estes sítios são chamados de *motivos* de reconhecimento e possuem tanto sua estrutura tridimensional quanto sua sequência de nucleotídeos conservada.

[1] Estrutura secundária de ácido nucleico que possui ao menos duas estruturas *stem-loop* em que metade de uma haste é intercalada entre as duas metades de outra haste.

3.3 Descoberta de *motivos*

A descoberta de *motivos* é um dos principais problemas da biologia molecular (24) e possui diversas aplicações, tais como:

- localização de sítios regulatórios em genes;
- identificação de redes de regulação gênica;
- identificação de sítios alvo para novas drogas;
- localização de sítios funcionais em proteínas;
- identificação da estrutura secundária em moléculas de RNA.

Existem várias técnicas experimentais que são utilizadas para determinar a localização desses padrões, tais como, *DNase footprinting* (42), *Gel-Shift* (43), *ChIP-seq* (44), *reporter construct assays* (45) e SELEX (46, 15).

O desafio consiste basicamente em encontrar padrões comuns em sequências biológicas de forma que, posteriormente, rótulos (funções) possam ser atribuídos a eles. Embora possuam precisão, técnicas experimentais são tipicamente caras e lentas. Por outro lado, abordagens computacionais vem sendo a bastante tempo aplicadas com sucesso em uma variedade de problemas biológicos e provaram sua eficiência.

Os *motivos* são sequências que possuem, em geral, de 5 à 30pb (pares de base) e são estatisticamente sobre-representados (do inglês *overrepresented*) (47). Em outras palavras, são padrões recorrentes passíveis de serem encontrados com a aplicação de técnicas matemáticas e algoritmos especializados.

A Descoberta de *motivo de novo* (do inglês *de novo motif discovery*) consiste em encontrar em um conjunto de dados, regiões que possuam alta similaridade. O problema é classificado como NP-difícil e tipicamente gera muitos falsos-positivos independentemente da metodologia utilizada (47). Isso significa que os estudos ainda carecem de ajustes e isso talvez seja a grande motivação que tem incentivado pesquisadores na criação de novas soluções.

3.3.1 Desafios

Devido a complexidade e ao grande número de padrões que podem ser encontrados, os algoritmos de busca tem apresentado baixa precisão. Estudos recentes mostraram

que, em média, a performance geral alcançada até o momento não passa dos 13% em sensitividade e 35% em precisão (47).

A descoberta de *motivos* é definida como um alinhamento múltiplo local, na qual assume-se que existe um alinhamento ótimo entre subsequências de um conjunto de dados que aplicado a uma função objetivo retornará um escore máximo (24). Existem várias instâncias deste problema:

- uma ocorrência por sequência: cada sequência possui apenas uma instância do *motivo*;
- zero ou uma ocorrência por sequência: algumas sequências podem não possuir um *motivo*;
- múltiplas ocorrências por sequência: algumas sequências podem possuir mais de uma instância de um mesmo *motivo*;
- múltiplos padrões: as sequências podem conter múltiplos *motivos* diferentes;

O número de falsos positivos é facilmente explicado utilizando um pouco de estatística. Assumindo que a probabilidade de fundo [2] de um determinado conjunto de dados seja uniforme, isto é, $P(A) = P(C) = P(G) = P(T) = 0.25$. É possível calcular qual a frequência esperada de um *motivo* no *dataset*. Por exemplo, o fator de transcrição *HindII* liga-se a um sítio cuja sequência consenso degenerada é GTYRAC (Y = *pYrimidine*. R = *puRine*). A probabilidade desta sequência é dada por

$$\left[\frac{1}{4}\right]^4 \times \left[\frac{1}{2}\right]^2 = 0.000976563$$

e ela aparecerá a cada $4^4 \times 2^2 = 1024$pb.

É importante notar que *in vitro*, o comportamento associado as moléculas pode ser diferente do encontrado *in vivo*. É provável que a maioria dos *motivos* encontrados *in silico* sem o devido tratamento, sejam biologicamente válidos *in vitro*. No entanto, além das afinidades físico-químicas que um fator de transcrição possui por seu sítio de ligação, experimentos *in vivo* demonstraram que outros elementos, como por exemplo, fatores fisiológicos, irão influenciar nesta ligação.

As sequências biológicas, principalmente em DNA, possuem elementos repetitivos e de baixa complexidade. Esses elementos tendem a atrapalhar na detecção de potenciais *motivos*. D'haeseleer (17) sugere a remoção destas regiões como uma etapa de

[2] Probabilidade que cada base apresenta dentro do genoma de seu respectivo organismo.

pré-processamento antes do inicio da busca. Com isso, o número de falsos positivos pode cair consideravelmente.

Existe uma forte correlação entre o número de falsos positivos, a quantidade e o comprimento das sequências. A. Zia e A. M. Moses (48), realizaram um estudo que apontou ser possível minimizar o número de falsos positivos realizando duas tarefas simples. A primeira consiste em escolher adequadamente o intervalo das sequências alvo, de modo a diminuí-las o máximo possível. A segunda resumi-se em adicionar mais sequências no *dataset*. Com isso a probabilidade em se encontrar um *motivo* aleatório sem função biológica tende a diminuir.

3.3.2 Principais metodologias para descoberta de *motivos*

Existem diversas metodologias que podem ser utilizadas na descoberta de *motivos*. Elas podem ser basicamente divididas em dois grupos: métodos exatos e métodos probabilísticos.

3.3.2.1 Métodos exatos

Os métodos por enumeração são algoritmos de busca exaustiva, isto é, percorrem todo o espaço de solução na busca pelo melhor *motivo* (17). Nesta categoria existem duas abordagens que se destacam: algoritmos baseados em palavras (WBA, do inglês *word-based algorithms*) e árvores de sufixo (ST, do inglês *suffix tree*) (17). WBA é uma abordagem que garante atingir o ótimo global sendo mais utilizado na busca por *motivos* pequenos (16). A ideia central do algoritmo é contar o número de ocorrências de todos os n-mers nas sequências alvo e calcular qual deles possui maior representação (17).

Os métodos baseados em WBA utilizam tipicamente a sequência consenso na construção dos *motivos*. Isso torna a sua representação muito rígida e pouco adequada para a maioria dos problemas biológicos reais. Como alternativa, as representações podem ser realizadas utilizando código degenerado IUPAC (49). Durante a fase de busca, os métodos baseados em WBA também tendem a gerar uma grande quantidade de falsos positivos, necessitando de um pós-processamento para remover os excessos (16).

O WBA pode se tornar uma estratégia de busca muito rápida quando implementado em conjunto com as árvores de sufixo. Van Helden et al. (50) desenvolveram um algoritmo utilizando ST que se mostrou eficaz na detecção de *motivos* no organismo *Saccharomyces cerevisiae*. No entanto, o algoritmo apresentou ineficiência na detecção de sítios de ligação grandes e pouco conservados.

Uma pesquisa similar foi conduzida por Sinha and Tompa (51). Eles desenvolveram

um algoritmo chamado YMF (*Yeast Motif Finder*), que utilizou ST juntamente com *Hidden Markov Models*. O estudo baseou-se na revalidação de sítios de ligação conhecidos do organismo *Saccharomyces cerevisiae*. O algoritmo foi capaz de revalidar 18 das 23 regiões reguladoras utilizadas no estudo.

Pavesi et al. (23) desenvolveram um algoritmo baseado em árvores de sufixo chamado Weeder que tentou aperfeiçoar a limitação imposta pelos métodos exatos. Eles estenderam a enumeração exaustiva também para padrões mais longos. No entanto, a abordagem foi capaz de encontrar *motivos* com no máximo 12pb.

3.3.2.2 Métodos probabilísticos

As abordagens probabilísticas possuem como principal característica a construção de modelos estatísticos capazes de representar um conjunto de sequências alvo. Os parâmetros do modelo, em geral, são estimados utilizando máxima verossimilhança ou inferência bayesiana (16).

Estes métodos normalmente exigem poucos parâmetros de configuração, no entanto, costumam ser bastante sensíveis aos dados que lhes são fornecidos como entrada (16). Eles tipicamente apresentam melhor desempenho para encontrar *motivos* longos e com menor nível de conservação.

Ao contrário das abordagens por enumeração, os métodos probabilísticos não varrem todo o espectro de soluções. Eles usam otimização local e heurísticas para acelerar a busca. Se por um lado isso torna a convergência mais rápida, por outro, pode levar o método a ótimos locais [3]. Alguns exemplos de algoritmos que implementam abordagens probabilísticas são: *Gibbs Sampling* (GS) e *Expectation Maximization* (EM).

Um dos primeiros trabalhos a utilizar algoritmos probabilísticos foi desenvolvido por Stormo e Hartzell (11), cuja finalidade consistiu em encontrar sítios com elevado nível de informação. Anos mais tarde esse mesmo algoritmo ganhou uma nova versão capaz de estimar a significância estatística de um determinado conjunto de dados utilizando *large deviation statistics* (10).

Lawrence et al. (52) desenvolveram um dos primeiros algoritmos baseados no método estatístico Gibbs Sampling capaz de encontrar um modelo otimizado de alinhamento múltiplo local. Isso permitiu a detecção e otimização de vários padrões e suas respectivas repetições. Este método foi aplicado com sucesso em proteínas *helix-turn-helix*, lipocalinas e preniltransferases.

[3] Solução ótima apenas dentre um conjunto de soluções vizinhas, porém, não-ótima dentre todas as soluções possíveis.

Outra abordagem que se destacou dentro dos métodos probabilísticos foi desenvolvido por Bailey et al. (53). O *Multiple EM for Motif Elicitation* (MEME) utiliza a heurística *Expectation-Maximization* para detectar *motivos* com baixo grau de conservação em sequências de biopolímeros não alinhadas. O objetivo do MEME consiste na descoberta de novos padrões em conjuntos de dados onde pouca ou nenhuma informação sobre quaisquer *motivos* são antecipadamente conhecidas.

3.3.2.3 Abordagens baseadas em computação evolutiva

Os métodos baseados em computação evolutiva podem ser vistos como abordagens híbridas de aprendizado de máquina que unem dentro de uma mesma estratégia conceitos de ambas classes citadas acima. Eles já foram aplicados em vários estudos anteriores de descoberta de *motivo*.

Um dos primeiros trabalhos foi desenvolvido por Corne et al. (54) e mostrou como regiões TSS-proximal poderiam ser representadas a partir de matrizes de peso e sequências consenso. Eles utilizaram um algoritmo genético simples para detectar e revalidar padrões em regiões promotoras de células eucarióticas do organismo *D. melanogaster*.

Fogel et al. (28) empregaram um modelo evolutivo de ilhas distribuídas para encontrar genes regulados pelo fatores de transcrição *octamer* e *kappa-B* (NF-kB, do inglês *nuclear factor kappa-light-chain-enhancer of activated B cells*). O algoritmo foi capaz de redescobrir sítios de ligação previamente conhecidos que haviam sido determinados experimentalmente bem como listas de TFBSs putativos ainda desconhecidos.

Che et al. (55) utilizaram um algoritmo genético canônico somado a uma estratégia de reposição e extinção de indivíduos pertencentes à população inicial. Eles conseguiram resultados satisfatórios no dataset *cyclic-AMP* (11) e em dados extraídos de experimentos de ChiP-chip do organismo *Saccharomyces cerevisiae*.

Chan et al. (14) utilizaram uma abordagem genética com filtragem local e técnicas de pós-processamento adaptativo na identificação de TFBS. O algoritmo combinou representações por posição e sequências consenso com um operador de filtragem para diminuir a quantidade de falsos positivos detectados durante o processo evolutivo. Eles também utilizaram pré-seleção para manter a diversidade e evitar ótimos locais. O pós-processamento com adição e remoção adaptativa foi desenvolvido para tratar casos gerais com números arbitrários de instâncias por sequência.

Congdon et al. (56) denotaram a competência de algoritmos evolutivos na localização de elementos reguladores em circunstâncias em que métodos exaustivos seriam intratáveis. Eles conseguiram revalidar *motivos* regulados pelos fatores de transcrição

octamer, *kappa*, CFTR (do inglês *Cystic fibrosis transmembrane conductance regulator*) e Citocromo P450.

Luo et al. (57) propuseram um algoritmo genético auto-imune que adotou mecanismos de regulação e vacina com o objetivo de inibir a degeneração dos anticorpos durante a evolução. Os resultados experimentais demonstraram a capacidade do método em encontrar *motivos* conhecidos em sequências de regiões promotoras relativamente longas e múltiplos *motivos* dentro de uma única execução.

Algoritmos evolutivos também foram aplicados ao problema da descoberta de motivos em sequências de aminoácidos. Estas e outras aplicações podem ser revisadas em (34). Para a maioria dessas abordagens, a ênfase se concentrou na aplicação de algoritmos evolutivos canônicos para resolver problemas de biossequência. A nossa motivação é ligeiramente diferente, na medida em que pretendemos utilizar a flexibilidade dos algoritmos evolutivos somados com a eficiência que algumas heurísticas possuem. Assim, foi possível desenvolver estratégias que são mais aplicáveis à resolução de problemas de descoberta de motivos em situações reais.

4 HEURÍSTICAS E META-HEURÍSTICAS

Algoritmos de otimização, em geral, enfrentam imensos problemas quando o objetivo é fornecer soluções exatas para problemas NP-difíceis. Problemas desta classe, independentemente de condições especiais ou propriedades particulares, exigem consumo de processamento exponenciais em tempo ou em espaço tornando o uso de abordagens exatas proibitivas. Diante disso, a pesquisa científica concentrou esforços no desenvolvimento e aperfeiçoamento de abordagens aproximativas. A essas estratégias foi dado o nome de heurísticas (58).

Problemas computacionais podem ser vistos como coleções de instâncias e suas respectivas soluções. Eles possuem uma entrada que, sem perda de generalidade, é codificada sobre o alfabeto $\{0, 1\}$. Em geral, o objetivo consiste em retornar como saída uma ou mais soluções que satisfaçam as condições diretas e indiretas de um determinado problema (59). Em outras palavras, um problema computacional é caracterizado pelas propriedades que a saída precisa satisfazer em relação à entrada. Eles podem ser classificados em (60):

- problemas de enumeração: encontre todas as soluções S que satisfaçam a condição C de um problema P;
- problemas de contagem: quantas soluções S que satisfazem a condição C o problema P possui?
- problemas de decisão: existe uma solução S que satisfaça a condição C no problema P?
- problemas de localização: encontre uma solução S que satisfaça a condição C no problema P;
- problemas de otimização: encontre uma solução S de forma a minimizar ou maximizar a função objetivo C respeitando as restrições impostas pelo problema P.

Uma instância de um problema de otimização combinatória é definida por um conjunto base finito parcialmente ordenado $E = \{1, \ldots, n\}$, um conjunto de soluções factíveis $F \subseteq 2^E$ e uma função de custo $f : 2^E \rightarrow \mathbb{R}$. No caso de um problema de minimização, onde se busca uma solução ótima global $S^* \in F$ tal que $f(S^*) \leq f(S), \forall S \in F$ que associa um valor real $f(S)$ a cada solução factível $S \in F$.

Um método heurístico é uma abordagem cujo objetivo consiste em encontrar soluções factíveis localmente ótimas (eventualmente globais) para um determinado problema computacional. Diferente da busca cega, a busca heurística usa informações do problema para direcionar o algoritmo a locais promissores do espaço de busca (61). Embora não existam conhecimentos ou provas matemáticas completas sobre seu funcionamento ou convergência, os métodos heurísticos, mesmo sem oferecer garantias, podem chegar muito próximo ou até mesmo resolver problemas complexos utilizando uma quantidade limitada de tempo e espaço. Um procedimento algorítmico desenvolvido através de um modelo cognitivo que utiliza regras baseadas na experiência pode ser considerado uma abordagem heurística (62).

A principal característica dos métodos heurísticos consiste em encontrar "boas" soluções em tempo computacional razoável bem como minimizar o espaço ocupado pelas configurações e estados que um determinado algoritmo pode assumir. Em particular, essas características são desejáveis quando o problema faz parte do sub-conjunto NP-Completo de problemas computacionais no que diz respeito ao espaço $S(n)$ e principalmente ao tempo $T(n)$. Abaixo uma definição de heurística segundo Rich and Knight (63):

> Para resolver eficientemente muitos problemas difíceis, geralmente é necessário comprometer as exigências de mobilidade e sistematicidade e construir uma estrutura de controle que não garanta encontrar a melhor resposta, mas que quase sempre encontre uma resposta muito boa ... a heurística é uma técnica que melhora a eficiência de um processo de busca, possivelmente sacrificando pretensões de completeza.

As heurísticas podem ser classificadas, em construtivas, de melhoramento e metaheurísticas. Uma heurística construtiva, consiste em tentar encontrar uma boa solução, considerando a cada iteração somente o próximo passo, ou seja, o critério de escolha é basicamente local. Ela parte de uma solução vazia e efetua sua construção inserindo sempre um partícula de cada vez, até formar uma solução completa. Algoritmos construtivos não possuem sistema de *backtracking*, ou seja, após inserir uma partícula, não é possível retirá-la da solução (64).

Por outro lado, as heurísticas de melhoramento atuam em solução factíveis já construídas. Uma solução é considerada factível se ela satisfaz todas as restrições presentes na formulação do problema. Desta forma, a heurística trabalha no melhoramento da solução atual, através da realização de sucessivos passos ou movimentos (65).

As meta-heurísticas são métodos globais de busca que combinam estratégias gerais e diretrizes de estrutura para adaptar métodos heurísticos específicos a um determinado tipo de problema (66). Elas são guiadas pelos seguintes princípios:

- utilizam informações coletadas em tempo real para guiar a busca para o ótimo global;
- possuem a capacidade de escapar de ótimos locais;
- examinam a estrutura das soluções vizinhas.

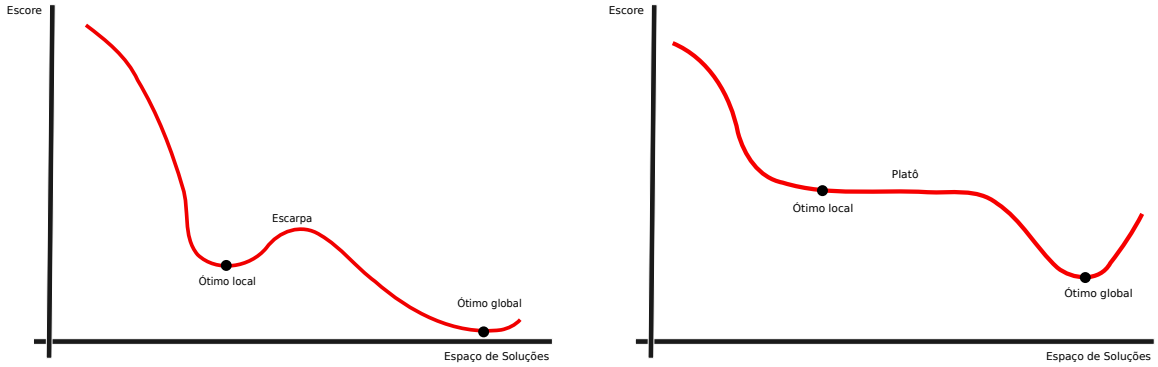
As meta-heurísticas são muito utilizadas em otimização combinatória e tendem a se mover rapidamente na direção de boas soluções. Por essa razão fornecem uma maneira muito eficiente de lidar com problemas grandes e de alta complexidade, normalmente intratáveis através dos métodos exatos. Ela também é eficiente em casos cujos os métodos tradicionais ficam presos em ótimos locais. Contudo, assim como as heurísticas tradicionais, as meta-heurísticas não garantem que a busca seja capaz de encontrar um ótimo global (67). Abaixo a definição de meta-heurística retirada da livro *Handbook of metaheuristics* (68):

> Uma meta-heurística é um conjunto de conceitos que pode ser utilizado para definir métodos heurísticos aplicáveis a um extenso conjunto de diferentes problemas. Em outras palavras, uma meta-heurística pode ser vista como uma estrutura algorítmica geral que pode ser aplicada a diferentes problemas de otimização com relativamente poucas modificações que possam adaptá-la a um problema específico. Alguns exemplos de metaheurísticas são: recozimento simulado, busca tabu, busca local iterativa, algoritmos evolutivos e otimização por colônia de formigas.

Além de permitir escapar de ótimos locais, as meta-heurísticas fornecem maneiras de resolver problemas provenientes da utilização de heurísticas tradicionais. Existem dois cenários bastante comuns que acontecem no emprego de técnicas exclusivamente heurísticas: a escarpa e o platô (69). O primeiro ocorre quando o método heurístico alcança um ótimo local. Neste ponto, não existe movimento capaz de encontrar uma solução vizinha com escore melhor que o atual, como ilustra a Figura 10a. Desta forma o algoritmo encerra a busca e retorna sua posição atual. O Segundo ocorre quando as soluções vizinhas possuem um custo estagnado. Em outras palavras, temos uma região plana que não produz melhoras por um determinado período de tempo como mostra a Figura 10b. Se forem adequadamente implementadas, as metaheurísticas podem mitigar estes problemas.

4.1 VNS

VNS, do inglês *Variable Neighborhood Search* (70), é uma meta-heurística desenvolvida por Nenad Mladenovic e Pierre Hansen baseada na exploração de múltiplas definições de vizinhança impostas ao mesmo espaço de solução. Cada uma das suas iterações possui duas etapas principais: agitação e busca local. Na fase de agitação, um vizinho da solução



(a) Escarpa (b) Platô

Figura 10 – Espaço de busca. Simulação de escarpa e platô.

atual é gerado aleatoriamente, em seguida uma busca local é aplicada à solução obtida pelo passo anterior. O VNS explora sistematicamente a ideia de mudança de ambiente, tanto na busca por ótimos locais quanto no processo que permite a ele escapar dos vales que os contêm (71).

O VNS é um método de busca local que explora o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. O algoritmo vai gradativamente explorando vizinhanças cada vez mais distantes e focaliza a busca em torno de uma nova solução somente se um movimento de melhora é realizado (72). O algoritmo 1 mostra o pseudocódigo da meta-heurística VNS.

O VNS básico pode ser definido em aproximadamente 4 passos:

Passo 1. seleciona-se um conjunto de estruturas de vizinhança $N_k = (k = 1, k = 2, \ldots, k_{max})$ e defini-se uma solução inicial x;

Passo 2. gera-se uma solução $x' \in N_k(x)$ aleatoriamente;

Passo 3. aplica-se busca local em x' com objetivo de encontrar um ótimo local x'';

Passo 4. se x'' for melhor que x, então x'' torna-se a solução atual, como mostra a Equação 4.1.

$$x = \begin{cases} x'', & \text{se } f(x'') < f(x) \\ x, & \text{caso contrário} \end{cases} \tag{4.1}$$

O algoritmo VNS tem como base as seguintes premissas:

Algoritmo 1: Pseudocódigo: VNS

Entrada: Solução factível x_0, número de estruturas de vizinhança r
Saída: Melhor solução encontrada

```
1  início
2      x* → x_0;
3      enquanto critério de parada não for satisfeito faça
4          k ← 1;
5          enquanto k ≤ r faça
6              x' → GERA_VIZINHO(x*, k) ;
7              x'' → BUSCA_LOCAL(x');
8              se f(x'') < f(x*) então
9                  x* → x'';
10                 k → 1;
11             fim
12             senão
13                 k → k + 1;
14             fim
15         fim
16     fim
17     retorna x*
18 fim
```

1. um mínimo local encontrado com o uso de uma determinada estrutura de vizinhança não é necessariamente um mínimo local para outra vizinhança;

2. o mínimo global é mínimo local para qualquer estrutura de vizinhança do problema, somente dependendo da solução inicial x_0;

3. em muitos casos, os mínimos locais podem estar próximos entre si.

A detecção do ótimo global, de acordo com a observação número três, pode ser guiada por buscas realizadas no entorno de bons ótimos locais. Desta forma, o VNS parte do princípio de que faz sentido aprofundar a busca em diferentes estruturas de vizinhanças para encontrar diferentes soluções localmente ótimas, e partir destas, encontrar o mínimo global. Supõe-se que um mínimo local foi alcançado na vizinhança selecionada quando melhorias na solução corrente não são mais encontradas pelo processo de busca. A Figura 11 ilustra o mecanismo de mudança de vizinhança do algoritmo VNS.

Nesta figura, é importante notar a característica do VNS em explorar diferentes vizinhanças. Na Figura 11a, o algoritmo encontra um melhoramento na vizinhança N_1. No entanto, caso isso não seja possível, o VNS busca por vizinhos melhores em vizinhanças mais distantes, como é o caso da Figura 11b. A Figura 11c é uma generalização que ilustra o algoritmo indo até níveis mais distantes à procura de soluções promissoras.

A arquitetura do VNS apesar de simples e possuir poucos parâmetros pode ser implementada utilizando organizações mais sofisticadas ou mesmo hibridizadas com outras meta-heurísticas. As principais variações são: busca em vizinhança variável descendente, busca em vizinhança variável clássica, busca em vizinhança variável descendente com pertubações, busca em vizinhança variável com oscilações.

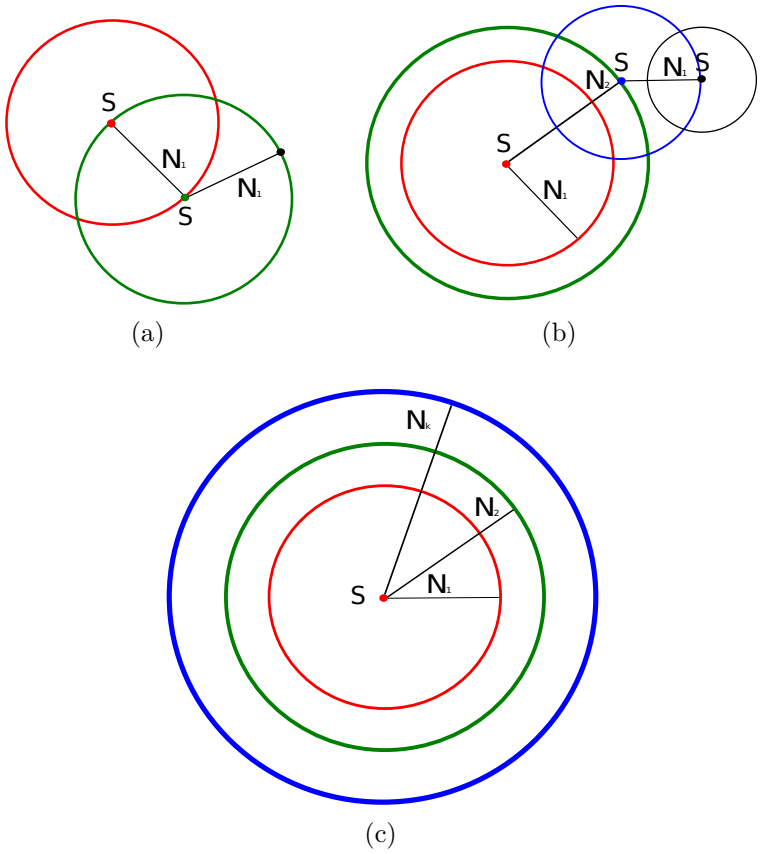


Figura 11 – Vizinhanças do algoritmo VNS. (a) Vizinhança de primeiro nível. (b) Vizinhança de segundo nível. (c) Vizinhança de nível k.

4.2 Grasp

GRASP, do inglês *Greedy Randomized Adaptive Search Procedures* (73), é uma meta-heurística criada por Thomas Feo e Mauricio Resende que implementa uma hibridação do algoritmo semi-guloso aliado a um método de busca local embutido em um framework multistart, no qual cada iteração abrange basicamente duas fases: construção e busca local. A fase de construção cria uma solução viável, cuja vizinhança é investigada até que um mínimo local seja encontrado durante a fase de busca local. A melhor solução global é mantida como resultado. GRASP e VNS podem ser utilizados de forma complementar, com a construção sendo aplicada pelo GRASP e a busca local pelo VNS (74).

A busca algorítmica estocástica tipicamente é caracterizada por duas etapas: (1) obtenção de configurações promissoras e factíveis do problema e (2) exploração das vizinhanças destas soluções na tentativa de melhorara-lás. Nem sempre é possível obter soluções factíveis e promissoras através da geração aleatória. Em virtude da explosão combinatória associada aos problemas NP-difíceis, existe um número proibitivo de configurações possíveis, tanto factíveis quanto infactíveis, que o problema pode apresentar. Isso exige grande esforço computacional na tarefa associada à etapa de melhoria.

A meta-heurística GRASP tenta solucionar o problema da diversidade que acompanha a busca gulosa ao mesmo tempo que permite a construção de soluções qualitativamente melhores quando comparado à inicialização aleatória. Um algoritmo semi-guloso, assim como o aleatório, não é capaz de gerar soluções factíveis em todos os casos. Quando isso acontece, um procedimento de reparo deve ser invocado para fazer mudanças na solução corrente de modo a torná-la factível (alternativamente, a solução pode ser simplesmente descartada e seguida por uma nova execução do algoritmo semi-guloso, até que uma solução factível seja construída) (75). O processo de busca local é aplicado em uma solução fornecida pelo algoritmo semi-guloso ou, se necessário, pelo procedimento de reparação. O algoritmo 6 ilustra o pseudocódigo do algoritmo GRASP.

Algoritmo 2: Pseudocódigo: Grasp

```
   Entrada: Instância do problema
   Saída: Melhor solução encontrada
 1 início
 2     x* ← ∞;
 3     enquanto critério de parada não for satisfeito faça
 4         x ← CONSTRUÇÃO();
 5         se x não é factível então
 6             x ← REPARO();
 7         fim
 8         x ← BUSCA_LOCAL();
 9         se f(x) < x* então
10             s* → x;
11             x* → f(x)
12         fim
13     fim
14     retorna s*
15 fim
```

A fase construtiva é executada através da estratégia gulosa aleatorizada e pode ser realizada através de diferentes abordagens. A mais conhecida é a técnica semi-gulosa de Hart e Shogan (76). Nessa estratégia, a escolha determinística foi substituída por um critério aleatório de triagem em um conjunto restrito organizado de maneira gulosa.

Em problemas de otimização, nem sempre o menor (ou maior) custo será o escolhido. As variáveis candidatas a serem incluídas na solução são organizadas em uma lista chamada de Lista Restrita de Candidatos ou LCR. O tamanho da LCR é tipicamente controlada por um parâmetro β fornecido na inicialização do algoritmo. No entanto, ela também pode ser controlada por um parâmetro $\alpha \in [0, 1]$, que varia entre duas condições extremas:

1. $\alpha = 0$: a LCR possuirá somente uma variável e a escolha torna-se gulosa;
2. $\alpha = 1$: a LCR englobará todos as variáveis e a escolha torna-se aleatória.

O controle do tamanho da LCR utilizando o parâmetro α pode ser implementado de acordo com a Equação 4.2:

$$LRC = \{e \in M | c(e) \leq c^{min} + \alpha \times (c^{max} - c^{min})\} \tag{4.2}$$

Onde e é o vértice a ser testado, M é o conjunto de vértices adjacentes, $c(e)$ é o custo do vértice a ser testado, c^{min} é o vértice adjacente de menor custo, α é o parâmetro que controla o tamanho da LCR e c^{max} é o vértice adjacente de maior custo.

Por exemplo, observando a Figura 12, se o tamanho da LCR for controlado por um parâmetro $\beta = 3$, então comporiam-a os nós 5, 4 e 3. Caso o controle fosse realizado pelo parâmetro $\alpha = 0.5$, então a decisão construtiva poderia ser realizada da seguinte forma:

$$LCR = \{e \in M | c(e) \leq 1 + 0.5 \times (4 - 1) \leq 2.5\}$$

Considerando o exemplo acima, apenas as soluções 4 e 5 possuem custo menor que 2.5 e por conseguinte comporiam a LCR.

Na fase de busca local, o GRASP pode empregar qualquer processo de busca heurística, no entanto, o algoritmo mais utilizado na literatura, inclusive o recomendado no artigo original, é o VNS. O VNS pode ser implementado de duas formas: *primeira melhora* ou *melhor melhora*. O método de *primeira melhora* é mais rápido enquanto que o método de *melhor melhora*, apesar de gerar soluções melhores, é mais lento e em alguns casos, a depender do tamanho do espaço de buca, pode ser computacionalmente inviável. Em aplicações reais, é frequente a utilização de ambas as estratégias durante a execução das iterações (75).

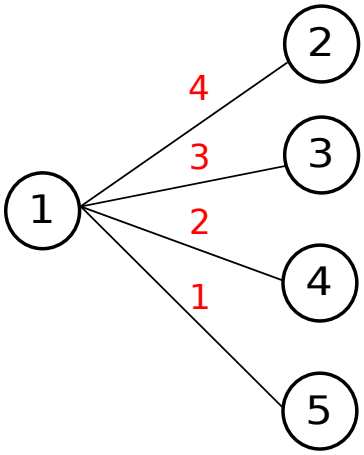


Figura 12 – Construção parcial de uma solução utilizando GRASP. O critério de escolha do próximo nó não é definido pelo custo da melhor aresta (aresta que liga o nó 1 ao nó 5, neste exemplo) e sim por uma estratégia semi-gulosa baseada em lista.

4.3 Recozimento simulado

A reprodução de sistemas naturais, também chamado de biomimética[1], tem contribuído e inspirado muitos tipos de algoritmos de otimização, principalmente àqueles que utilizam programação inteira (58).

O Recozimento Simulado (SA, do inglês *Simulated Annealing*) é uma meta-heurística probabilística – baseada no método Monte Carlo e consequentemente no algoritmo Metropolis (77) – que segue esta linha, cujo conceito foi inspirado no sistema físico da termodinâmica (78). A técnica consiste em aquecer metais a altas temperaturas e permitir seu arrefecimento de forma lenta o suficiente para que seus átomos consigam se organizar em um estado de energia miníma. Com isso o material ganha rigidez e consistência (79).

Comparando o SA computacional com a abordagem física, temos: (1) os estados possíveis de um metal correspondem a soluções do espaço de busca; (2) a energia em cada estado físico corresponde ao valor da função objetivo; e (3) a energia mínima (se o problema for de minimização ou máxima, se for de maximização) corresponde ao valor de uma solução ótima local, possivelmente global.

Sem perda de generalidade e considerando problemas de minimização, o conceito básico do algoritmo consiste em gerar a cada iteração um novo estado a partir do estado corrente de forma aleatória. Se o novo estado possuir menor energia (função objetivo) em relação ao estado anterior, então, o novo estado passa a ser o estado corrente (79). Porém, se o novo estado possuir maior energia, a probabilidade de transição do estado corrente

[1] Área da ciência que tem por objetivo o estudo de suas estruturas e funções, objetivando aprender com a natureza e não sobre ela.

para o novo é definido pela Equação 4.3 (80):

$$P_{i,j} = \exp\left(-\frac{|\Delta|}{T_i}\right) \tag{4.3}$$

Onde:

- $P_{i,j}$ é a probabilidade do algoritmo transitar do estado i para o estado j;
- Δ é a variação do custo da função;
- T_i é a temperatura no instante i.

A Equação 4.3 define o controle da transição entre os estados possíveis do sistema quando $\Delta \geq 0$. Esse controle é realizado através de uma simplificação da distribuição de probabilidades proposta por Maxwell-Boltzmann, cuja proposta segue a ideia geral do algoritmo Metropolis-Hastings. Um número R é uniformemente sorteado e comparado ao resultado da Equação 4.3, onde espera-se o seguinte comportamento:

$$S_i = \begin{cases} S_j, & \text{se } R \leq P_{i,j} \\ S_i, & \text{caso contrário} \end{cases} \tag{4.4}$$

O parâmetro Δ equivale ao valor da variação da função objetivo decorrente da troca da solução S_i pela S_j, valor que também pode ser associado à variação de energia entre os estados E_i e o estado E_{i+1} do sistema termodinâmico: $\Delta = f(S_j) - f(S_i) = E_{i+1} - E_i$. Assim, se:

- $\Delta < 0$: Existe uma redução de energia associada à transição. Considerando-se um problema de minimização, tal transição estaria à descoberta de uma configuração melhor que a corrente. Desta forma, o algoritmo sempre aceitaria transitar da solução S_i para S_j;
- $\Delta = 0$: Não há alteração de energia e aceitação da solução seria indiferente;
- $\Delta \geq 0$: Há um aumento do estado de energia. A aceitação deste tipo de solução deve ser controla pelo método.

No inicio das iterações, o SA permite que movimentos “ruins” sejam realizados. Em outras palavras, em altas temperaturas, cada estado possui praticamente as mesmas

probabilidades de serem escolhidos como estado corrente. Atingido-se o equilíbrio térmico, a temperatura é atualizada (Equação 4.5) e somente os estados de menor energia dispõem de alta probabilidade para se tornarem os estados correntes. Na pratica, com baixas temperaturas, o SA se comporta como o algoritmo Descida de Encosta. Desta forma, temos que, para $T \to \infty$ então $P_{i,j} = 1$ (busca aleatória) e para $T \to 0$ então $P_{i,j} = 0$ (busca gulosa).

$$T_i = \alpha \times T_{i-1} \tag{4.5}$$

Onde:

- T_i representa a temperatura atual do sistema;
- T_{i-1} é a temperatura do sistema no estado anterior;
- α é o fator de decremento da temperatura. $0 < \alpha < 1$.

A Equação 4.5 mostra o decremento geométrico da temperatura proposto por Kirkpatrick et al. (78). No entanto, outras maneiras de decremento também são possíveis, da quais vale ressaltar: decremento linear, decremento de Lundy e Mees (81), decremento de Aarts e Krost (79), dentre outros. O algoritmo 3 exibe o pseudocódigo SA.

4.4 Computação Evolutiva

Os Algoritmos Evolutivos (EA, do inglês *Evolutionary Algorithms*) podem ser descritos como um conjunto genérico e adaptável(82) de técnicas computacionais que empregam conceitos provenientes dos processos naturais evolutivos com o objetivo de resolver problemas de otimização (83). Eles possuem como principal característica a estocasticidade, isto é, são processos fortemente influenciados pela teoria das probabilidades.

Embora possam existir vários subtipos de EAs, todos eles apresentam alguns aspectos em comum, tais como: uso de populações de indivíduos, recombinação de características, mutação e seleção dos indivíduos mais aptos. É possível fazer uma breve analogia com os processos biológicos, onde a adaptação ao meio ambiente representa o problema a ser otimizado e cada ser vivo interpreta uma possível solução computacional, que sofre frequentes pressões evolutivas por parte do ambiente (problema). Da mesma forma que alguns organismos vivos possuem maior adaptabilidade do que outros para sobreviverem em determinados ambientes, algumas soluções computacionais também possuem níveis de

Algoritmo 3: Pseudocódigo: Recozimento Simulado

Entrada: Instância do problema, T_k, L_k
Saída: Melhor solução encontrada

```
1  início
2      x* ← x_k;
3      f* ← f(x_k);
4      enquanto tratamento térmico estiver ativo faça
5          enquanto regra de parada for falsa faça
6              para i ← 1 até L_k faça
7                  gerar nova solução x_{k+1} ∈ Q(x_k);
8                  calcular Δ = f(x_{k+1}) − f(x_k);
9                  se Δ ≤ 0 então
10                     x_k ← x_{k+1};
11                     se f(x_k) < f* então
12                         x* ← x_k;
13                         f* ← f(x_k)
14                     fim
15                 fim
16                 senão
17                     se rand[0,1] < exp(−Δ/T_k) então
18                         x_k ← x_{k+1};
19                     fim
20                 fim
21             fim
22             atualiza(L_k);
23             atualiza(T_k);
24         fim
25         reconfigura(L_k);
26         reconfigura(T_k);
27     fim
28 fim
```

adaptação maiores do que outras na resolução de um problema específico (82). Segundo De Jong, (84), uma abordagem evolutiva deve possuir:

- uma ou mais populações de indivíduos competindo por limitados recursos;
- mudanças na constituição da população governadas pelo nascimento e morte de indivíduos;
- conceitos de fitness que reflete a capacidade de um indivíduo sobreviver e se adaptar ao ambiente;
- conceitos de hereditariedade, onde a prole se assemelha, mas, não é idêntica a seus pais.

Uma característica positiva que a computação evolutiva apresenta em relação às demais, consiste em seu alto poder de abstração. Desta forma, é possível aplicá-la nos mais variados tipos de problemas com poucas modificações na estrutura geral do algoritmo. A Computação Evolutiva originou-se basicamente a partir de três grandes linhas de pesquisa: Programação Evolutiva (EP, do inglês *Evolutionary Programming*), Estratégias Evolutivas (ES, do inglês *Evolutionary Strategy*) e os Algoritmos Genéticos (GA, do inglês *Genetic Algorithm*) (85). Além disso, ela pode ser classificada em: Computação Evolutiva Mono-Objetivo e Computação Evolutiva Multiobjetivo. O algoritmo 4 ilustra um EA básico.

Algoritmo 4: Pseudocódigo: Algoritmo Evolutivo básico

Entrada: Tamanho da população, taxa de recombinação, taxa de mutação
Saída: População final de soluções

```
1  início
2      Inicializa população de forma aleatória;
3      Avalia indivíduos (calcula fitness);
4      enquanto critério de parada não for satisfeito faça
5          P^i ← SELECIONA melhores indivíduos;
6          P^ii ← RECOMBINA indivíduos de acordo com taxa de recombinação;
7          P^iii ← MUTACIONA de acordo com taxa de mutação;
8          AVALIA indivíduos (P^iii);
9      fim
10 fim
```

4.4.1 Nomenclatura

Os algoritmos evolutivos foram inspirados em características biológicas de cunho evolutivo e por essa razão, sua arquitetura possui alguns nomes extraídos da biologia, como:

- cromossomos, genes e alelos;
- seleção natural e *Fitness*;
- reprodução (sexuada e assexuada) e hereditariedade;
- elitismo;
- mutação.

4.4.1.1 Cromossomos, genes e alelos

Um indivíduo ou cromossomo, em termos computacionais, pode ser definido como uma estrutura de dados que representa uma possível solução para o problema (7). Os

indivíduos, em geral, são representados por *strings*, números binários, números contínuos, números discretos ou uma mistura destes. O conteúdo de um cromossomo, representa um determinado genótipo que é expresso na forma de fenótipo (7). Em outras palavras, o conteúdo de um cromossomo que armazena as medidas de uma casa está para o genótipo assim como a casa já construída está para o fenótipo.

Um "cromossomo computacional", assim como o biológico, é formado por genes. Uma estrutura de dados bastante utilizada para representação dos cromossomos é o vetor. Quando um cromossomo é codificado utilizando-se dessa estrutura, os genes correspondem aos índices e os valores que cada índice assume constitui um alelo. Denomina-se *locus*, uma posição específica de um gene dentro de um vetor e uma coleção de cromossomos é chamada de população (7). A Figura 13, mostra de forma resumida todas as estruturas discutidas neste paragrafo.

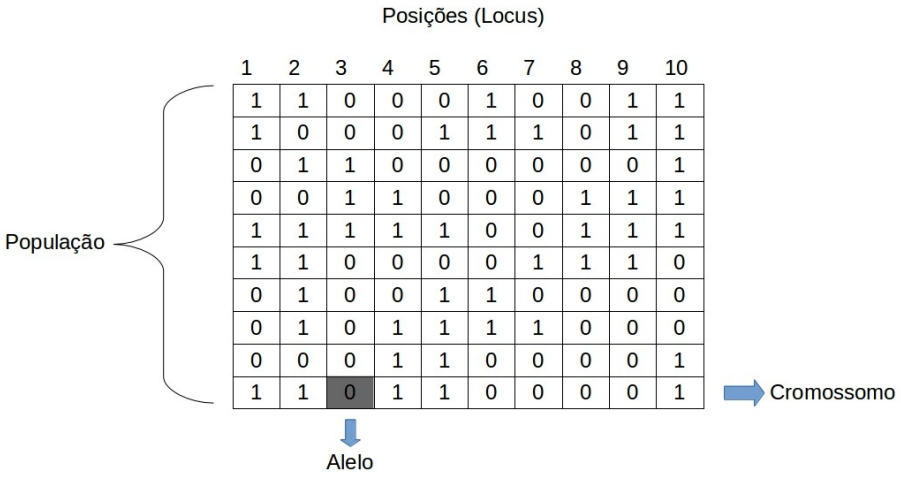


Figura 13 – Exemplo de uma matriz 10x10 representando uma população de indivíduos com codificação binária. Fonte: (7).

4.4.1.2 Seleção natural e Fitness

Embora a biologia evolutiva possua um conceito de *fitness* um pouco divergente do adotado nos algoritmos evolutivos, é possível sintetizar que, ambos, fornecem parâmetros ou medidas que podem ser utilizadas como métricas para mensurar a capacidade de adaptação de um indivíduo ao meio (84).

A Função *Fitness* é a propriedade que confere aos algoritmos evolutivos estas característica. Em problemas de otimização, ela é utilizada como um fator de custo, que provê um escore e direciona a seleção dos indivíduos mais aptos que irão compor a próxima geração (7).

Os métodos de seleção mais utilizados pelos algoritmos evolutivos são: roleta e tor-

neio. As Equação 4.6 e Equação 4.7, representam essas duas abordagens, respectivamente.

$$P_k = \frac{f(V_k)}{\sum_i f(V_i)} \tag{4.6}$$

Onde $f(V_k)$ é o escore do indivíduo k, $\sum_i f(V_i)$ é o somatório dos escores de todos os indivíduos pertencentes a população e P_k é a probabilidade do indivíduo k fazer parte da próxima geração.

$$P = \max[f(V_i), f(V_j)], \quad 0 \leqslant i, j \leqslant L \quad e \quad i \neq j \tag{4.7}$$

Onde P é o vetor que receberá os indivíduos vencedores do torneio realizado entre os membros da população, $f(V_i)$ e $f(V_j)$ representam o escore calculado dos elementos i e j respectivamente e L representa o número de elementos da população atual. É interessante notar que a Equação 4.7 mostra o torneio entre dois indivíduos de uma mesma população, no entanto, está abordagem pode ser expandida para além de dois participantes.

4.4.1.3 Reprodução e hereditariedade

Assim como na biologia, os algoritmos evolutivos possuem conceitos de recombinação e hereditariedade. O objetivo em ambos é basicamente o mesmo: trocar material genético entre os diversos indivíduos de uma população para que, com o passar do tempo, possam surgir indivíduos melhores e mais adaptados. A recombinação confere ao algoritmo evolutivo a capacidade de explotação, i.e, da intensificação da busca dentro do espaço de soluções (84).

As algoritmos evolutivos permitem o uso de diversos tipos e operadores de recombinação, porém, o mais primordial é chamado de *Single Point Crossover* (Figura 14). Ele consiste em sortear aleatoriamente um "ponto de corte" e recombinar partes dos indivíduos pais para formarem os indivíduos filhos. Um exemplo de abordagem encontrado em (7) consiste em colocar na geração subsequente apenas os filhos que possuírem um fitness maior que o fitness médio da população atual.

4.4.1.4 Elitismo

Embora a definição de elitismo seja simples, suas implicações na evolução do algoritmo são de extrema importância. O elitismo é definido como uma ação evolucionária que busca preservar os indivíduos mais aptos ao meio. Em outras palavras, o algoritmo busca preservar as melhores soluções, incorporando-as na geração subsequente. Desta forma, evita-se que indivíduos com alto grau de adaptabilidade sejam descartados no

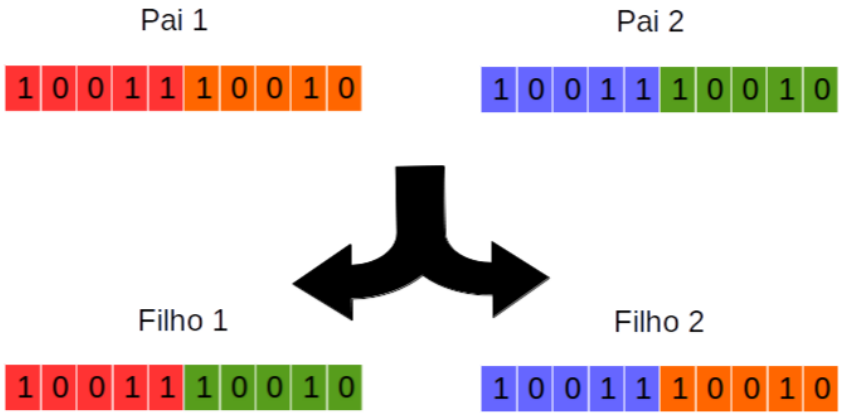


Figura 14 – Recombinação de um ponto entre indivíduos de uma população.

processo evolucionário. A implicação direta da aplicação deste conceito é uma convergência mais rápida, no entanto, isso pode levar o algoritmo atingir máximos ou mínimos locais (84).

Para evitar ótimos locais, os algoritmos, em geral, guardam apenas a melhor solução de cada geração. Com isso, o algoritmo ganha capacidade para explorar vários "picos" ou "vales" sem o problema da convergência prematura (84). Em problemas onde existem vários ótimos locais, abordagens não-elitistas costumam ser mais eficientes, no entanto, a convergência pode se torna lenta (84). Uma característica que a implementação do elitismo confere aos algoritmos evolutivos é a monotonicidade do gráfico da sua função de convergência.

4.4.1.5 Mutação

A mutação é um evento pontual que ocorre em indivíduos de uma população, tanto computacionais quanto biológicos, que pode alterar (ou não) o genótipo, o comportamento, a função ou o fenótipo dos indivíduos afetados. Seu objetivo consiste em realizar a exploração do espaço de busca de forma mais rápida e eficiente.

Assumindo que um cromossomo computacional foi desenvolvido utilizando uma estrutura de dados do tipo vetor, uma mutação consiste na mudança do valor de um gene, em geral, escolhido ao acaso. Uma distribuição uniforme de probabilidade pode ser utilizada para definir qual valor será atribuído ao gene alvo (84). Este tipo de mutação é conhecido como *Random Mutation Operator* (Equação 4.8).

$$M_U(x) = x + U([a, b]) \tag{4.8}$$

Onde $U([a, b])$ denota a uma distribuição uniforme dentro do intervalo de $[a, b]$.

Do mesmo modo que existem vários operadores de recombinação, também existem diversos operadores de mutação, cada qual com características e desempenhos próprios que são utilizados em situações específicas. Além do *Random Mutation Operator*, citado acima, o *Gaussian Mutation Operator* é bastante utilizado em representações envolvendo números reais. Ele consiste em mudar os valores dos genes utilizando uma distribuição de probabilidade gaussiana Equação 4.9:

$$M_G(x) = x + N(0, \sigma) \tag{4.9}$$

Onde $N(0, \sigma)$ é um vetor de variáveis aleatórias gaussianas independentes, com média zero e desvio padrão σ.

4.5 Algoritmos Meméticos

Os Algoritmos Meméticos (60) (MAs, do inglês *Memetic Algorithms*) surgiram no final dos anos 80 para denotar uma família de meta-heurísticas que possuíam conceitos fortemente separados, como por exemplo os Algoritmos Evolutivos e a Recozimento Simulado. O adjetivo "memético" vem do termo "meme", cunhado por R. Dawkins (86) para denotar um análogo ao gene no contexto da evolução cultural. Citando Dawkins:

> *Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain via a process which, in the broad sense, can be called imitation.*

A citação acima ilustra a filosofia central dos MAs: melhoria individual somada a cooperação populacional. Como foi o caso para os EAs clássicos, os MAs tiveram inicialmente baixa aceitação, porém, agora estão se tornando cada vez mais populares.

Algoritmos meméticos possuem elementos de metaheurísticas e inteligência computacional. Embora possuam princípios de Algoritmos Evolutivos, eles não são considerados estritamente evolutivos. Algoritmos Meméticos têm semelhanças funcionais com os Algoritmos Evolucionários Baldwinianos e Lamarckianos, Algoritmos Evolutivos Híbridos e Algoritmos Culturais. Uma definição elegante diz que um MA é uma estratégia de busca na qual uma população de agentes otimizadores cooperam sinergicamente e competem entre si (87).

As técnicas que os MAs podem utilizar na exploração dos problemas incluem, incorporação de heurísticas, utilização de algoritmos de aproximação, emprego de técnicas de busca local, aplicação de operadores de recombinação especializados, métodos exatos truncados, dentre outros. Além disso, um fator importante é o uso de representações adequadas do problema que está sendo abordado (88).

Assim como os EAs, os MAs são metaheurísticas populacionais. Isto significa que o algoritmo mantém uma população de soluções para o problema em questão, isto é, um conjunto de componentes que compreende várias soluções simultaneamente. Cada indivíduo da população representa uma possível solução para o problema. Estas soluções estão sujeitas a processos de competição e cooperação mútua de uma forma bastante similar aos padrões comportamentais de seres vivos de uma mesma espécie. O conceito dos MAs sugere que a cada geração uma atualização ocorra na população de indivíduos de forma que isso leve o algoritmo a melhores soluções para o problema que está sendo enfrentado. Há três componentes principais neste passo geracional: seleção, reprodução e substituição (89).

A etapa de seleção é a responsável (em conjunto com a etapa de substituição) pelos aspectos de competição entre indivíduos da população. Usando as informações fornecidas pela função de aptidão, o fitness dos indivíduos da população é avaliado. Posteriormente, uma amostra de indivíduos é selecionada para reprodução de acordo com esta medida de desempenho (90).

A substituição está fortemente relacionada ao aspecto de competição entre os indivíduos. Este componente cuida da manutenção da população em um tamanho constante. Para isso, os indivíduos mais velhos da população são substituídos pelos recém-criados (obtidos a partir da reprodução) utilizando algum critério específico. Isso pode ser feito unindo os melhores indivíduos da população inicial e intermediária (técnica chamada de estratégia da substituição da soma), ou simplesmente escolhendo os melhores indivíduos de população intermediária e inserindo-os em na população inicial substituindo os piores (91).

Talvez o aspecto mais interessante neste processo de geração seja a fase intermediária da reprodução. Nesta fase, são criados novos indivíduos a partir dos existentes. Isto é feito através da aplicação dos operadores reprodutivos. Embora muitos operadores diferentes possam ser utilizados, a situação mais típica envolve o uso de apenas dois operadores: recombinação e mutação.

Assim como os algoritmos evolutivos, este operadores são responsáveis pela explotação e exploração do espaço de busca. No entanto, nos AEs, o operador de mutação deve

gerar uma nova solução modificando parcialmente uma solução existente. Essa modificação pode ser aleatória - como normalmente é o caso - ou pode ser dotada de informações dependentes do problema de modo a polarizar a busca para regiões provavelmente boas do espaço de soluções. É precisamente à luz desta última possibilidade que se introduz um dos componentes mais distintivos dos MAs: os otimizadores locais (89).

Para compreender sua filosofia, vamos considerar a seguinte formulação abstrata: um operador de mutação que realiza uma modificação aleatória mínima em uma solução. Considere agora um grafo cujos vértices são soluções e cujas arestas conectam os pares de vértices de modo que as soluções correspondentes possam ser obtidas através da aplicação do operador de mutação em um deles. Um otimizador local é um processo que começa em um determinado vértice e se move para um vértice adjacente, desde que a solução vizinha seja melhor que a solução atual (92). Neste caso, a distância entre os vértices (comprimento da aresta) é determinado por meio de um função de terminação. Um exemplo simples consiste na terminação do percurso quando não são mais possíveis movimentos ascendentes (ou descendentes). No entanto, outras formas de terminação podem ser empregadas. Por exemplo, pode ser atribuído ao caminho um comprimento máximo permitido ou o término do algoritmo pode ocorrer assim que uma melhoria suficientemente boa no valor do fitness ser alcançada.

O otimizador local é utilizado em diferentes partes do algoritmo, pois, ele se trata de um operador. Por exemplo, pode ser inserido após a utilização de qualquer outro operador de recombinação ou mutação. Alternativamente, poderia ser empregado apenas no final do estágio reprodutivo. Assim, os MAs podem ser vistos como uma coleção de agentes (indivíduos) realizando uma exploração autônoma do espaço de busca, cooperando algumas vezes via recombinação, e competindo por recursos computacionais devido ao uso de mecanismos de seleção / substituição (93).

Este modelo geral possui algumas características importantes. Em primeiro lugar, o processo responsável pela geração da população inicial pode ser realizado de forma aleatória. Em outros casos, aplica-se um mecanismo de semeadura mais sofisticado (por exemplo, alguma heurística construtiva), através do qual soluções de alta qualidade são injetadas na população inicial.

Existe um outro elemento que deve ser levado em consideração no processo de geração da população inicial: a técnica de auto-reinicio (94). Este processo é muito importante no que diz respeito ao uso apropriado dos recursos computacionais. Considere que a população pode chegar a um estado em que a geração de nova solução melhorada é muito improvável. Este poderia ser o caso quando todos os agentes da população são muito semelhantes entre si. Nessa situação, o algoritmo provavelmente gastará a maior

parte do tempo em pontos de reamostragem numa região muito limitada do espaço de busca, com o subsequente desperdício de esforços computacionais (95).

Esse fenômeno é conhecido como convergência, e pode ser identificado usando medidas como a entropia de Shannon (96). Se esta medida cai abaixo de um limiar predefinido, a população é considerada em um estado degenerado. Este limiar depende da representação do problema que está sendo tratado e, portanto, deve ser determinado de forma empírica. Uma possibilidade alternativa é utilizar uma abordagem probabilística para determinar com uma determinada confiança o nível de convergência da população. Por exemplo, em (97) uma abordagem Bayesiana é apresentada para este propósito.

Uma vez que a população é considerada em um estado degenerado, o processo de reinício é invocado. Novamente, isso pode ser implementado de várias maneiras. Uma estratégia típica é manter uma fração da população atual (esta fração pode ser tão pequena quanto uma solução, a melhor atual) e substituir os indivíduos remanescentes por soluções recém-geradas.

5 ABORDAGEM PROPOSTA

Neste capítulo apresentamos a abordagem proposta MFMD (do Inglês, *Memetic Framework for Motif Discovery*). Esta abordagem é uma evolução das abordagens DMEC, do inglês *Discovery Motifs by Evolutionary Computation* (98) e DMMA, do inglês *Discovery Motifs by Memetic Algorithms* (99). Primeiramente, iremos descrever brevemente as abordagens DMEC e DMMA, e consequentemente daremos foco na abordagem MFMD. Maiores detalhes sobre DMEC e DMMA podem ser revisados em (98, 99).

5.1 DMEC

A proposta do DMEC consiste na evolução de uma população de matrizes PSSM [1], do inglês *Position Specific Scoring Matrix*, utilizando a estrutura de um algoritmo evolutivo canônico com um operador de mutação guloso. Foram obtidos bons resultados em vários datasets sintéticos e alguns reais, como por exemplo, o dataset cyclic-AMP (CRP) (11).

A população inicial foi gerada aleatoriamente com representação vetorial inteira. Cada cromossomo pode ser visto como um alinhamento local múltiplo, onde seus genes armazenam as posições iniciais de cada *motivo* possível. Um valor w, que se refere ao tamanho do *motivo* a ser encontrado, deve ser passado como um parâmetro para o algoritmo.

A avaliação de cada indivíduo foi calculada utilizando a entropia relativa ou divergência de *Kullback-Leibler*. A entropia relativa é uma medição que indica o quanto um *motivo* desvia-se da distribuição de nucleotídeos de fundo. Após a inicialização aleatória de cromossomos, cada indivíduo é transformado em uma subsequência de tamanho w. Eles são posicionados um abaixo do outro, formando uma matriz $n \times w$, onde n é o número de sequências disponíveis no conjunto de dados e w é o tamanho do *motivo* a ser descoberto. Após o cálculo da aptidão, a seleção é realizada usando a técnica do torneio, onde dois indivíduos são selecionados aleatoriamente, e suas pontuações são comparadas. O cromossomo com maior escore ganha, e consequentemente é selecionado para compor a próxima geração da população.

[1] Matriz de pesos criada a partir de um alinhamento múltiplo local de sequências e é comumente utilizada para representar modelos probabilísticos de motivos.

5.2 DMMA

O DMMA consiste em uma evolução do DMEC, onde foram consideradas algumas heurísticas juntamente com a estrutura clássica dos algoritmos evolutivos. Ainda nesta abordagem, foram utilizados os algoritmos *Simulated Annealing* (SA) e o *Variable Neigborhood Search* (VNS) como heurísticas de busca. O SA foi utilizado para melhorar as soluções iniciais geradas randomicamente enquanto que o VNS foi aplicado nos operadores de seleção e de mutação respectivamente. Com essas alterações o algoritmo DMMA obteve um ganho substancial em relação ao seu antecessor DMEC.

Assim como no DMEC, o problema é representado por um vetor de inteiros, onde cada índice corresponde à respectiva sequência a partir do qual o *motivo* foi extraído. Além disso, os valores assumidos pelos índices descrevem a posição inicial de cada solução candidata. A população inicial P foi construída de forma aleatória e, em seguida, uma busca local é realizada utilizando o algoritmo *Simulated Annealing* com o objetivo de descobrir indivíduos promitentes e direcionar a busca para regiões mais promissoras.

O cálculo do fitness foi realizado convertendo as posições iniciais de cada solução em uma matriz PSSM. O fitness foi calculado utilizando a abordagem bi-objectivo de Soma de Pesos Ponderada, cujas funções utilizadas foram: *Information Content score* e *Complexity Score*. O MFMD surgiu como uma evolução do DMMA e a principal diferença entre eles está na forma como o MFMD constrói e organiza a população inicial.

5.3 MFMD

O MFMD foi desenvolvido utilizando linguagem de programação Java release 8u111 (64 bits) e sistema operacional Linux Ubuntu 14.04 LTS. A ideia central do arcabouço consiste em evoluir uma população de matrizes PSSM e encontrar soluções que maximizem o *Information Content Score* e o *Complexity Score* utilizando uma abordagem bi-objetivo de Soma de Pesos Ponderada [2]. Para isso, o MFMD foi dividido em três etapas: Pré-Processamento, Descoberta de Padrão e Correspondência de Padrão.

O principal estágio do MFMD é o de Descoberta de Padrão, onde um algoritmo baseado na meta-heurística GRASP foi utilizado na inicialização da população. Uma estrutura de dados do tipo árvore também foi empregada no armazenamento das soluções, o que permitiu maior flexibilidade e eficiência na construção do algoritmo.

[2] A soma de pesos só é realizada caso a etapa de pré-processamento não seja executada. Maiores detalhes na subseção 5.3.2.2

5.3.1 Pré-processamento

Esta etapa visa encontrar e remover dos conjuntos de dados sub-sequências que possam direcionar a busca a locais inválidos. Segundo D'haeseleer (17) essas sub-sequências, chamadas de espúrios (100), podem contribuir negativamente com o desempenho dos algoritmos de busca. Ele sugere algumas orientações com o objetivo de mitigar este problema, das quais vale ressaltar:

1. Remover subsequências espúrias, pouco complexas e com baixo nível informação;
2. Remover elementos repetitivos do dataset, que possam direcionar a busca a locais impróprios.

Antes do algoritmo iniciar a fase de Descoberta de Padrões, dois programas são executados com a finalidade de atender aos requisitos acima. O primeiro é o DUST (101), uma ferramenta criada por R. L. Tatusov and D. J. Lipman cuja finalidade consiste em remover do dataset sub-sequências com baixa complexidade. O segundo é o RepeatMasker (102), criado por A. Smit, R. Hubley and P. Green, que se destina a remover padrões repetitivos de sequências.

5.3.2 Descoberta de padrão

Esta fase consiste na otimização e descoberta da melhor matriz PSSM a partir de um conjunto de dados de entrada. Para alcançar este objetivo, este ciclo foi sub-dividido cinco sub-etapas: Construção da População Inicial, Cálculo do Fitness, Recombinação, Mutação e Seleção. O algoritmo 5 ilustra como essa fase é executada. É importante notar que o critério de parada adotado neste algoritmo é o número máximo de iterações.

5.3.2.1 População inicial

Cada solução é representada por uma estrutura de dados do tipo árvore. Nesta estrutura, os nós representam as posições iniciais, sendo que o nó raiz representa a posição inicial da primeira sequência do dataset. Deste modo o algoritmo cria uma árvore de soluções para cada posição válida do dataset de sequências. Por exemplo, caso o dataset possua 100 posições válidas, então o algoritmo gerará 100 árvores, cada qual com sua respectiva posição inicial. O cálculo do número total de posições válidas pode ser obtido pela Equação $v = L - w + 1$, onde v é o número total de posição válidas, L é o tamanho de cada sequência e w é o tamanho de um particular *motivo*. No MFMD, as soluções são construídas gradativamente com o auxílio de uma heurística baseada em GRASP. Isso contribuiu com a condução das soluções iniciais a locais mais promissores do espaço de busca.

Algoritmo 5: Pseudocódigo: Descoberta de Padrão

Entrada: Dataset
Saída: A melhor matriz PSSM encontrada

```
1  início
2      P ← inicializa a população P com o algoritmo baseado em Grasp;
3      Q ← inicializa a população Q vazia;
4      R ← inicializa a população R vazia;
5      enquanto critério de parada não for satisfeito faça
6          para i = 1 to nr_recomb faça
7              S ← um subconjunto de P;
8              Escolha S_1, S_2 ∈ P aleatoriamente;
9              Max ← max(f(S_1), f(S_2));
10             Filho ← recombinação(S_1, S_2);
11             se Max ≥ f(Filho) então
12                 Filho ← busca local com VNS;
13             fim
14             Q ← Filho;
15         fim
16         R ← P ∪ Q;
17         P ← n_pop melhores soluções pertencentes a R;
18     fim
19 fim
```

Em trabalhos anteriores, Andronescu e Rastegari (103) desenvolveram uma abordagem de descoberta de *motivos* utilizando a implementação original do GRASP que pode ser revisada em (73). Para este trabalho, no entanto, foram realizadas algumas modificações no GRASP original a fim de otimizá-lo e adaptá-lo ao problema de descoberta de *motivos*. Estas alterações se mostraram essenciais, pois, na sua forma original, o GRASP obteve resultados inferiores quando comparado ao algoritmo aqui apresentado.

As modificações consistiram no emprego de uma variável q que modifica o comportamento do algoritmo e determina se o mesmo fará uma escolha gulosa ou aleatória. Por se tratar de um algoritmo memético, a função multi-start também foi desabilitada porque nesta abordagem o GRASP é utilizado apenas como ferramenta de inicialização. Então, a cada iteração, um número $n \in [0, 1]$ é uniformemente sorteado e o comportamento do algoritmo segue a Equação 5.1:

$$escolha = \begin{cases} gulosa, & n \leq q. \\ aleatoria, & \text{caso contrário.} \end{cases} \tag{5.1}$$

Caso a escolha seja gulosa, o algoritmo ainda testa se existem outras posições que possuem escore igual ao melhor escore encontrado até o momento, isto é, se existe empate

entre os escores da lista de posições válidas. Em caso positivo, todas as posições empatadas são adicionadas à árvore. Caso a escolha não seja gulosa, as soluções são ranqueadas em uma Lista de Candidatos Restritos (LCR). Em seguida uma solução desta lista é uniformemente escolhida e adicionada na árvore.

O tamanho da LCR e do parâmetro q podem variar, porém, em nossos testes constatamos empiricamente que $LCR = 5$ e $q = 0.9$ reproduziram os melhores resultados. O algoritmo termina quando todas as posições iniciais forem adicionadas à árvore, isto é, quando a altura da árvore for igual ao número de sequências do dataset menos um.

A complexidade do algoritmo cresce de acordo com o tamanho do dataset. Por exemplo, se um dataset possui N sequências de tamanho $L = 30$ e *motivos* com comprimento $w = 5$ existirão $L - w + 1 = 26$ posições válidas neste dataset. Sendo assim, o algoritmo fará 26^2 comparações entre a primeira e a segunda sequência, mais 26^2 comparações entre a segunda e a terceira sequência e assim por diante. Desta forma, a complexidade final do algoritmo é $O((L - w + 1)^2 \times N - 1)$ que pode ser resumida em $O(N \times L^2)$. No pior caso o algoritmo pode alcançar a complexidade $O(L^N)$ se todas as posições válidas de todas as sequências do dataset empatarem nos termos de valor de escore. No entanto isso é extremamente improvável e na prática temos apenas alguns empates ocorrendo a cada iteração, sendo que, em média, a complexidade $O(N \times L^2)$ prevalece.

A Figura 15 ilustra como a geração da população inicial é realizada. Após a construção de todas as árvores, as soluções são ordenadas de forma descendente. Em seguida as 3 melhores soluções de cada árvore são escolhidas para compor a população inicial. Por exemplo, em um dataset com 30 posições válidas, a população inicial possuirá um total de 90 indivíduos. Valores inferiores e superiores a 3 foram testados, no entanto, não surtiram efeitos positivos nos resultados, sendo que 3 apresentou um bom compromisso entre velocidade e precisão. A Figura 16 mostra uma árvore de soluções gerada a partir de um dataset com 4 sequências cuja raiz é a posição 60 e o algoritmo 6 apresenta como esta etapa é executada.

5.3.2.2 Cálculo do fitness

O fitness é calculado realizando a conversão das posições iniciais de cada indivíduo em uma estrutura denominada Alinhamento Múltiplo Local de Sequências (MLSA, do inglês *Multiple Local Sequence Alignment*). A partir do MLSA é possível calcular a Matriz de Escore de Posição Específica (PSSM, do inglês *position-specific scoring matrix*). A PSSM é uma matriz de markov não homogênea de ordem zero (18), tipicamente empregada na construção de modelos probabilísticos de *motivos* cuja independência estatística entre as diferentes "colunas" do alinhamento é assumida. Isso significa que, do ponto de vista estatístico, as bases nucleicas que formam os elementos regulatórios não possuem correlação

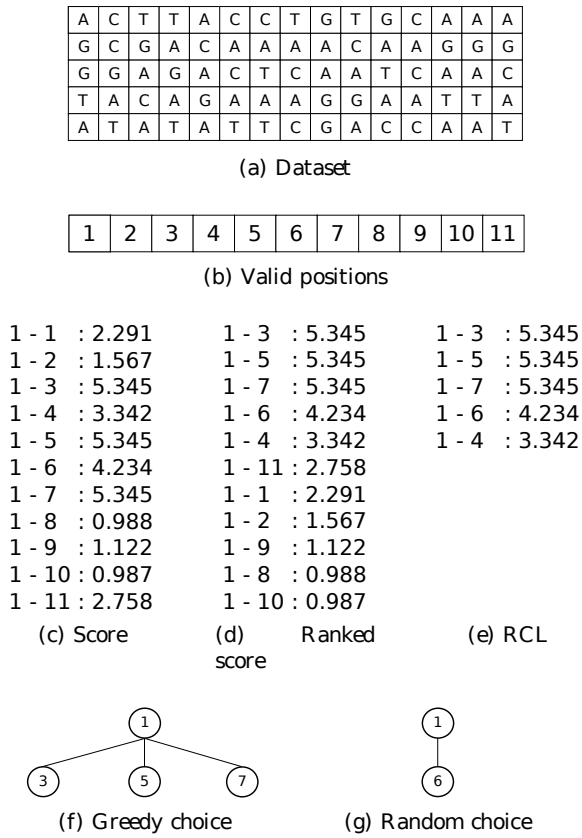


Figura 15 – (a) Representação do dataset de sequências. (b) Posições válidas ($w = 5$). (c) Cálculo do escore das posições válidas (11 no exemplo). (d) Classificação descendente da lista de escore. (e) Lista de candidatos restrito com tamanho 5. (f) Construção da árvore pela escolha gulosa. Todas as posições que possuem escores iguais são adicionadas à árvore. (g) Construção da árvore pela LCR. Neste caso a escolha é unitária e uma solução é escolhida randomicamente da LCR para compor o próximo nível da árvore.

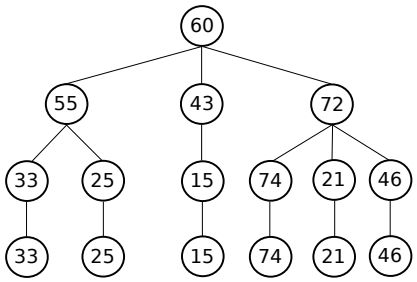


Figura 16 – Árvore de soluções. O nó raiz representa a posição inicial da primeira sequência do dataset. O caminho da raiz ate cada nó folha configura uma solução. O total de soluções que uma árvore apresenta é igual ao número de nós folha que ela possui.

entre si. Na prática, segundo Benos et al., essa independência é uma boa aproximação (104). Portanto a probabilidade de uma matriz PSSM pode ser determinada por uma

Algoritmo 6: Inicialização da população

```
Entrada: Dataset
Saída: Conjunto de soluções iniciais
início
    V ← soluções válidas;
    A ← árvores;
    para i = 1 to |V| faça
        A[i] ← insere raiz com as posições de V;
    fim
    para j = 1 to |A| faça
        se n ≤ q então
            A[j] ← adiciona nó(s) de forma gulosa;
        senão
            A[j] ← adiciona nó a partir da LCR;
    fim
fim
```

distribuição multinomial, como definida pela Equação 5.2:

$$P = \prod_{j=1}^{w} \left[\frac{N!}{\prod_{i=1}^{A} n_{i,j}!} \prod_{i=1}^{A} p_i^{n_{i,j}} \right] \tag{5.2}$$

Onde i refere-se as linhas da matriz, j refere-se as colunas, A é o tamanho do alfabeto utilizado (4 para nucleotídeos), w é o tamanho do *motivo*, p_i é a probabilidade a priori do nucleotídeo i (pode ser inferida através da análise do genoma do indivíduo), $n_{i,j}$ é o número de ocorrências do nucleotídeo i na posição j e N é o número total de sequências presentes no alinhamento.

Para um *motivo* de tamanho w, uma PSSM assume a forma de matriz $4 \times w$. A Figura 17 ilustra todas as etapas pertinentes à construção desta matriz. Maiores detalhes podem ser revisados em (8).

O fitness de cada indivíduo foi calculado através da abordagem bi-objetivo soma de pesos ponderada, cujas funções utilizadas foram: *Information Content* (IC) (Equação 5.3) e *Complexity Score* (CS) (Equação 5.4).

$$\mathrm{IC} = \sum_{i=1}^{\Sigma} \sum_{j=1}^{w} \Theta_{(i,j)} \log_2 \left[\frac{\Theta_{(i,j)}}{\Theta_{(0,i)}} \right] \tag{5.3}$$

Onde w é o tamanho do *motivo*, Σ é o número de letras pertencentes ao alfabeto. ($\Sigma = 4$ para nucleotídeos), $\Theta_{(i,j)}$ é a matriz das frequências relativas, $\Theta_{(0,i)}$ é o vetor das

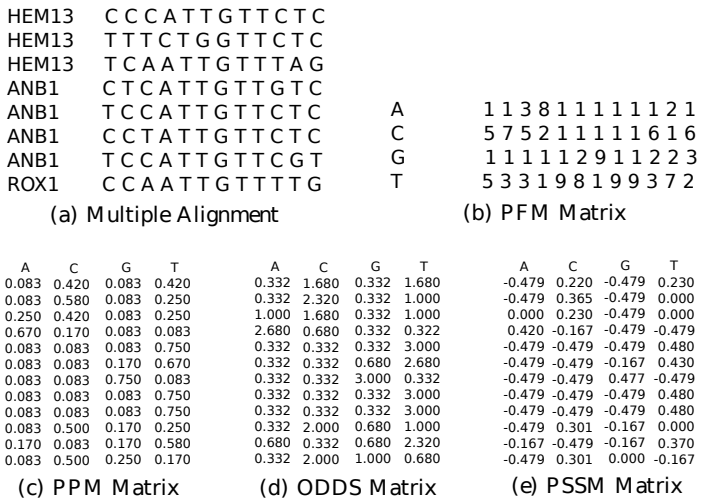


Figura 17 – (a) Oito conhecidos sítios de ligação em três genes de *S. cerevisiae*. (b) Matriz PFM considerando os pseudocontadores. (c) Matriz de frequência relativa de nucleotídeos (PPM). (d) Divisão de cada índice da matriz PPM pela probabilidade de fundo (PODDS). (e) Logaritmo natural de cada índice da Matriz POODS (PSSM).

probabilidades de fundo.

$$\mathrm{CS} = \log_N \left[\frac{w!}{\Pi n_i} \right] \tag{5.4}$$

Onde $N = 4$ para nucleotídeos, w é o tamanho do *motivo*, n_i é o número total de nucleotídeos $i \in A, C, G, T$.

O IC pode ser interpretado como uma estimativa da energia que um conjunto de *motivos* exerce sobre seu respectivo sítio de ligação em oposição ao resto do genoma do organismo (15). Em outras palavras, o IC mede a diferença estatística entre um *motivo* pertencer a um modelo probabilístico específico ou pertencer a um modelo probabilístico de fundo (frequentemente inferido através das sequências genômicas de um determinado organismo). O modelo estatístico de fundo é tipicamente construído sob uma cadeia de markov homogênea de ordem zero ou superior.

Já o *Complexity Score* foi definido por Gary B. Fogel e Weekes (28) e penaliza sequências com baixa complexidade, isto é, sequências cujo valor da entropia é muito baixo. Em geral, isto pode atrapalhar a busca e deve ser considerado um ruído (48). Por exemplo, o *motivo* "aaaaaa" ($n_a = 6, n_c = 0, n_g = 0, n_t = 0$) possuirá uma complexidade miníma uma vez que obterá um valor máximo em Πn_i. O *motivo* "atacgt" ($n_a = 2, n_c = 1, n_g = 1, n_t = 2$) obterá um valor de complexidade maior em relação ao anterior, uma vez que o valor da funcão Πn_i será menor. Neste exemplo, $CS(aaaaaa) = \frac{6!}{6\times6\times6\times6\times6\times6} = \frac{720}{46656} = 0.0154$ e

$CS(atacgt) = \frac{6!}{2\times2\times2\times1\times1\times2} = \frac{720}{16} = 45.$

Desde que o DUST seja executado na etapa de pre-processamento, o MFMD pode ser reduzido a um algoritmo mono-objetivo executando apenas o *Information Content Score*. Isso traz como beneficio uma execução mais rápida e não compromete a acurácia da abordagem.

Quando os fatores de transcrição se comportam como dímeros, o complemento reverso do *motivo* também deve ser levado em consideração. Caso o *motivo* seja palíndromo, essa predileção pode levar o algoritmo a resultados mais acurados. É importante notar que ao inserir o complemento reverso no cálculo do escore, a matriz PSSM se transforma em uma matriz simétrica.

Desta forma, o fitness total de cada indivíduo fica definido pela Equação 5.5:

$$Fi = v_1 \times IC + v_2 \times CS \tag{5.5}$$

Onde: v_1 e $v_2 \in [0, 1]$ são pesos escolhidos de maneira arbitrária. Em nossos testes, $v1 = 0.8$ e $v2 = 0.2$ foram os valores que reproduziram os melhores resultados.

5.3.2.3 Recombinação e mutação

Na recombinação, uma parcela da população inicial P é escolhida ao acaso para participar do processo. Em seguida a recombinação ocorre entre pares desta lista e os indivíduos filhos são colocados dentro de uma população intermediária Q que possui três vezes o tamanho de P. A cada recombinação, o algoritmo calcula o escore dos pais p_1 e p_2, seleciona o maior e o coloca em p^*. Após os filhos c_1 e c_2 serem gerados, o escore destes também são calculados e comparados com p^*. Se $F(c_1) < p^*$ então a mutação ocorre através da busca local em c_1 utilizando a heurística VNS. A mesma situação vale para o filho c_2.

É importante notar que o cruzamento entre os pais foi realizado através da recombinação com um ponto de corte. Outros tipos de cruzamento foram implementados, no entanto, não foram capazes de melhorar o compromisso entre velocidade e precisão. A razão do êxito deste operador se deve ao fato de existir uma forte relação de vizinhança entre as soluções candidatas. Desta forma, o operador de *crossover* foi capaz de realizar a intensificação da busca sem destruir ou comprometer os melhores *schemas* encontrados.

5.3.2.4 Seleção

Após a mutação as populações P e Q são unidas, formando a população R ($R = P \cup Q$). Em seguida, a população R é ordenada de forma descendente e as $|P|$ primeiras soluções de R são copiadas novamente para a população P. É importante destacar que a escolha dos indivíduos em R é realizada de maneira gulosa, sendo assim, o mecanismo de elitismo estś sendo utizado nesta etapa.

5.3.3 Correspondência de padrão

Esta etapa consiste na aplicação de técnicas estatísticas para o reconhecimento de *motivos* que não foram encontrados durante a etapa de Descoberta de Padrão. Em muitos casos, as regiões promotoras possuem mais de um sítio de ligação. Por conseguinte, é esperado que os algoritmos de busca sejam capazes de encontrar o maior número possível de *motivos* pertencentes a um determinado gene co-regulado.

Para isso nós dividimos o dataset em janelas de tamanho w ou w-mers e calculamos o p-valor de cada janela utilizando a matriz PSSM encontrada na fase de Descoberta de Padrão. O p-valor é a probabilidade de se encontrar uma pontuação maior ou igual à observada de forma aleatória (105). As regiões cujo p-valor calculado são menores que um ponto de corte previamente estabelecido são classificadas como possíveis *motivos*.

O MFMD assume que a distribuição final dos escores é uma distribuição gaussiana (106) de média μ e desvio padrão σ ($X \sim N(\mu, \sigma^2)$). Os parâmetros do modelo estatístico foram estimados utilizando os escores previamente calculados com o auxílio da matriz PSSM encontrada na etapa anterior. Desta forma, os scores são normalizados e transformados em z-scores utilizando a Equação 5.6:

$$z = \frac{x - \mu}{\sigma} \tag{5.6}$$

Onde x é o escore bruto, μ é a média e σ é o desvio padrão estimados.

Em seguida o p-valor é calculado utilizando a função distribuição acumulada definida pela Equação 5.7:

$$F_X(x) = \int_{-\infty}^{x} f_X(t) \;\; dt \tag{5.7}$$

Onde $F_X(x) = P(X \leq x)$ ou $P(a < X < b) = F_X(b) - F_X(a)$, onde $b = 1$ e $a = x$.

O objetivo é calcular a área abaixo da curva e encontrar quais posições possuem maior significância estatística como mostra o gráfico da Figura 18:

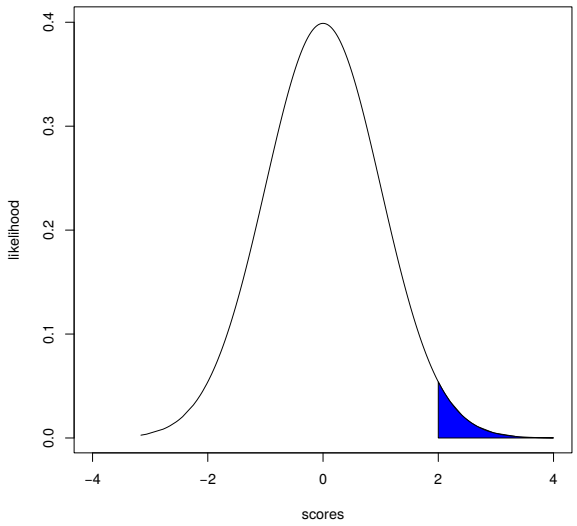


Figura 18 – Curva plotada a partir do cálculo dos escores. A área marcada em azul ilustra de forma simulada os p-valores que estariam abaixo do nível de significância (ex. 0.0001) estabelecido pelo usuário. Portanto, essa seria a área de interesse, onde estariam localizados o escores que apresentam maior força de ligação entre os *motivos* e o respectivo fator que os regula.

Em suma, os seguintes passos são executados nesta etapa (Figura 19):

1. Dividir todo o dataset em fragmentos de tamanho w;
2. Calcular a probabilidade de cada fragmento utilizando o modelo probabilístico encontrado na etapa de Descoberta de Padrão, i.e, calcular a $Pr(seq|Modelo)$;
3. Normalizar os scores e transformá-los em z-scores;
4. Calcular a função distribuição acumulada (FDA) para cada z-score;
5. Escolher somente os valores que satisfazem o nível de significância (ex. 0.0001) previamente estabelecido pelo usuário.

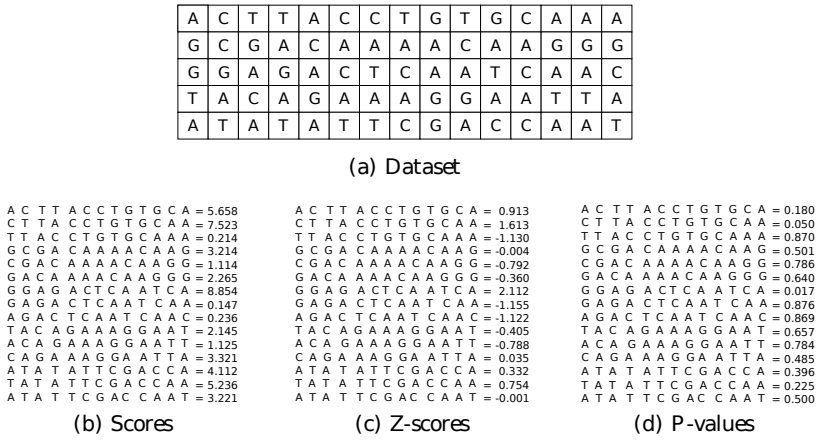


Figura 19 – (a) Dataset de sequências. (b) Divisão do dataset em w-mers ($w = 13$). Para cada janela, o escore é calculado utilizando a matriz PSSM encontrada na etapa de Descoberta de Padrão. (c) Transformação dos escores em z-scores. (d) Os p-valores são calculados a partir dos z-scores. Um ponto de corte pode ser utilizado para classificar novos *motivos*.

6 RESULTADOS

Para demonstrar a eficiência do MFMD foram realizados vários experimentos concentrados em 4 grupos de datasets:

- sintéticos: datasets e *motivos* gerados algoritmicamente;
- semi-sintéticos: datasets gerados algoritmicamente e *motivos* extraídos de sítios reais;
- ChIP–seq: datasets e *motivos* extraídos de experimentos de ChIP–seq;
- reais: datasets e *motivos* extraídos de sítios reais.

É importante notar que tanto os experimentos reais quanto os experimentos ChIP-seq foram realizados com datasets e *motivos* reais. Para cada dataset, 30 experimentos foram executados e os resultados obtidos foram comparados entre as abordagens MFMD, DMMA, *Gibbs Motif Sampler* (107) e MEME (do inglês, *Multiple EM for Motif Elicitation*) (19).

O MEME foi desenvolvido por *Bailey* e *Elkan* e utiliza uma estratégia baseada no algoritmo *Expectation-Maximization* (EM) para inferir as posições iniciais dos *motivos*. Na etapa de *Expectation*, o algoritmo calcula a matriz de frequência de cada posição válida, cujos valores são utilizados na etapa de *Maximization* para determinar a localização de cada sub-sequência que melhor se ajusta ao modelo.

O *Gibbs Motif Sampler* foi desenvolvido por *Rouchka* e *Thompson* e utiliza um método similar ao EM cuja finalidade consiste na descoberta do padrão mais provável por amostragem estatística com o objetivo de maximizar a relação entre modelo probabilístico específico e o modelo probabilístico de fundo.

Para comparar o desempenho de cada programa, foram utilizadas métricas padrão de recuperação de informações: *precision* e *recall* [1] (108). Sete medidas básicas de desempenho foram calculadas:

- verdadeiro positivo (TP): número de posições iniciais corretas que cada algoritmo conseguiu encontrar;

[1] A tradução deste termos para o português seriam coeficientes de revocação (recall) e precisão (precision). No entanto, por padronização, resolvemos deixar as denominações em inglês.

- falso positivo (FP): número de posições iniciais incorretas preditas por cada algoritmo;
- falso negativo (FN): número de posições iniciais corretas que cada algoritmo não conseguiu encontrar;
- verdadeiro negativo (TN): número de posições iniciais incorretas não preditas por cada algoritmo;
- *precision* $\frac{TP}{TP+FP}$: mede a proporção das posições iniciais corretamente preditas dentre as positivas preditas;
- *recall* $\frac{TP}{TP+FN}$: mede a proporção das posições iniciais corretamente preditas dentre as positivas verdadeiras;
- *f-score* $(2 \times \frac{precision \times recall}{precision+recall})$: teste de acurácia que calcula a média harmônica entre *precision* e *recall*, tendo como melhor valor 1 e pior 0.

Para medir a performance de cada estratégia, o critério adotado foi a posição inicial que cada abordagem conseguiu encontrar. Uma posição é considerada correta se ela for igual à real ou variar duas unidades para mais ou para menos. Por exemplo, se posição anotada de um determinado *motivo* for 60, todos estes valores serão considerados corretos: 58, 59, 60, 61 e 62. Para cada experimento realizado com o MFMD e DMMA a média e o desvio padrão das medidas de desempenho foram calculadas. Para os experimentos efetuados com o MEME e *Gibbs Motif Sampler* foram utilizados os valores cuja execução apresentou melhor performance.

6.1 Datasets utilizados

6.1.1 Sintéticos

Foram gerados 300 datasets sintéticos com sequências possuindo 500pb cada divididos em grupos de acordo com 3 fatores: tamanho do dataset, tamanho do *motivo* e nível de conservação do *motivo*. Foram criados 6 diferentes grupos que misturam estes três fatores (ilustrados na Tabela 1) com 50 datasets cada um. A probabilidade de fundo utilizada seguiu a seguinte distribuição: $A\left(\frac{1}{10}\right)$, $C\left(\frac{2}{10}\right)$, $G\left(\frac{3}{10}\right)$ e $T\left(\frac{4}{10}\right)$.

6.1.2 Semi-sintéticos

Foram gerados 11 conjuntos de dados sintéticos com sequência possuindo 500pb cada contendo *motivos* reais extraídos do sítio Jaspar (109). Os datasets que compõem esta experiência foram escolhidos aleatoriamente com base na sua correspondente identificação. A Tabela 2 ilustra as principais propriedades deste conjunto de dados.

Tabela 1 – Resumo dos datasets sintéticos.

Grupo	Numero de sequências	Tamanho do *motivo*	Nível de conservação
1	100	20	90%
2	100	20	50%
3	100	10	90%
4	30	20	90%
5	30	20	50%
6	30	10	90%

Tabela 2 – Resumo dos datasets semi-sintéticos.

ID	Nome	Espécie	Quantidade de *motivos*
MA0165.1	ABD-B	*D. melanogaster*	21
MA0004.1	ARNT	*M. musculus*	20
MA0096.1	BZIP910	*A. majus*	35
MA0172.1	CG11294	*D. melanogaster*	15
MA0260.1	CHE–1	*C. elegans*	27
MA0135.1	LHK3	*M. musculus*	20
MA0069.1	PAX6	*H. sapiens*	43
MA0107.1	RELA	*H. sapiens*	18
MA0086.1	SNA	*D. melanogaster*	40
MA0119.1	TLX1	*H. sapiens*	16
MA0016.1	USP	*D. melanogaster*	38

6.1.3 ChIP–seq

Tal como nos experimentos semi-sintéticos, os datasets escolhidos para integrar os ensaios ChIP-seq foram aleatoriamente selecionados com base na sua identificação. Foram utilizados 5 conjuntos de dados extraídos do sítio Jaspar com sequências possuindo 114pb cada construídos com elementos coletados a partir de experimentos de ChIP–seq. A Tabela 3 exibe as principais características destes datasets.

Tabela 3 – Resumo dos datasets ChIP–seq.

ID	Nome	Especie	Quantidade de sequências
MA0003.2	TFAP2A	H. sapiens	5098
MA0036.2	GATA2	H. sapiens	4380
MA0037.2	GATA3	H. sapiens	4628
MA0050.2	IRF1	H. sapiens	1362
MA0150.2	NFE2L2	M. musculus	726

6.1.4 Reais

Foram utilizados onze datasets no total, sete extraídos do sítio ABS (110), três extraídos do sítio SCPD (111) e um extraído da publicação de Stormo e Hartzell (11).

O fator de transcrição CREB, do inglês *cAMP Response Element-Binding Protein*, se liga a determinadas sequências de DNA chamadas elementos de resposta ao cAMP (112). Foram analisadas 17 sequências de 501pb cada contendo um total de 19 sítios de ligação. Genes cuja transcrição é regulada pelo CREB incluem: c-fos, BDNF, tirosina hidroxilase, neuropeptídeos e genes envolvidos no relógio circadiano de mamíferos (113).

A proteína CRP, do inglês *cAMP Receptor Protein* , é um fator de transcrição do organismo *E. coli* (112). Foram analisadas 18 sequências, cada uma com 105pb, contendo 24 sítios de ligação no total. O CRP é responsável por ativar a transcrição através de interações proteína-proteína com RNA polimerase (114).

O fator de transcrição HNF-1, do inglês *Hepatocyte Nuclear Factor-1*, é expresso em órgãos de origem endoderma, incluindo fígado, rins, pâncreas, intestinos, estômago, baço, timo, testículos e queratinócitos e melanócitos na pele humana (115). Neste dataset foram analisadas 22 sequências com 501pb e 27 sítios de ligação no total.

As proteíinas MEF2, do inglês *Myocyte Enhancer Factor-2*, são uma família de fatores de transcrição importantes no processo de diferenciação celular e desempenham um papel crítico no desenvolvimento embrionário (116). Este dataset possui 17 sequências com 501pb e 17 *motivos*. O MEF2 foi inicialmente identificado como um complexo fator de transcrição através da análise da região promotora do gene Creatina Quinase Muscular (CQM) com o objetivo de identificar fatores nucleares que interagem com a região de potenciador de CQM durante a diferenciação muscular (117).

MyoD, do inglês *Myogenic Differentiation-1*, é uma proteína que desempenha um papel importante na regulação da diferenciação muscular. O dataset MyoD possui 17 sequências com 501pb e 21 sítios de ligação no total. O gene MyoD é expresso em níveis extremamente baixos e essencialmente indetectáveis em células satélites quiescentes e a expressão de MyoD pode ser ativada em resposta a exercícios ou danos nos tecidos musculares (118).

NF-kB, do inglês *Nuclear Factor Kappa-Light-Chain-Enhancer of Activated B Cells*, é um complexo proteico que controla a transcrição do DNA, produção de citocinas e sobrevivência celular. O dataset NF-kB possui 6 sequências com 501pb e 8 motifs no total. Ele desempenha um papel fundamental na regulação da resposta imune à infecções e a sua incorreta regulação tem sido associada ao câncer, doenças inflamatórias e auto-imunes,

choque séptico, infecção viral e desenvolvimento imune inadequado (119).

O fator de resposta sérica (SRF, do inglês *Serum Response Factor*) é um membro da superfamília MADS de fatores de transcrição (120). Esta proteína pode ligar-se ao elemento de resposta ao soro (ERS) na região promotora de genes alvo. Este dataset possui 20 sequências com 501pb cada e 36 sítios de ligação no total. A proteína SRF é importante durante o desenvolvimento do embrião, uma vez que tem sido associada à formação da mesoderma. Nos mamíferos, ela é crucial para o crescimento do músculo esquelético (121).

A proteína de ligação TATA (TBP, do inglês *TATA-Binding Protein*) é um fator de transcrição geral que se liga especificamente a uma sequência de DNA denominada sítio TATA (122). O dataset TBP possui 95 sequências com 501pb cada e um total de 95 *motivos*. Em geral estes sítios são encontrados a cerca de 30 pares de bases a montante do local de início da transcrição em alguns promotores de genes eucarióticos (123).

A proteína PDR3 é um fator de transcrição que regula a resposta do fármaco pleiotrópico em S. cerevisiae. Ela pode apresentar-se na forma de homodímeros ou heterodímeros (124). Foram analisadas 7 sequências de 550pb cada que contêm 18 sítios de ligação. Essa proteína pode agir tanto como ativadora como repressora da atividade transcricional através da ligação a elementos pleiotrópicos de resposta a fármacos (PDREs) presentes nos promotores de genes alvo envolvidos na resistência a múltiplos fármacos (125).

A proteína REB1 é um fator de transcrição que se liga a uma sequência específica do DNA e age como um intensificador para a RNA polimerase I. O dataset REB1 apresenta 15 sequências com 105pb cada e possui 20 *motivos* no total. Ela pode se ligar a genes transcritos tanto pela RNA polimerase I como pela RNA polimerase II e é necessária para o término da transcrição da RNA polimerase I (126).

No organismo *S. cerevisiae*, o compromisso de entrar no ciclo celular mitótico ocorre durante no final da fase G1 em um ponto chamado *Start* (127). A ativação transcricional de genes neste ponto é largamente dependente de dois fatores de transcrição heterodiméricos denominados SBF e MBF. O complexo de MBF (fator de ligação à MCB) reconhece o elemento MCB e ativa a transcrição específica dos genes de ciclina e muitos outros necessários para a síntese de DNA, incluindo CDC9, POL1 , RNR1 e CDC21 (128, 129). O dataset MCB apresenta 6 sequências com 550pb cada e possui um total 12 *motivos*.

6.2 Análise dos resultados

As abordagens foram avaliadas de acordo com as métricas de recuperação de informação *precision*, *recall* e *f-score*. Essas medidas possuem como valor mínimo **0** e valor máximo **1**, sendo que **0** representa nenhuma posição predita e **1** caracteriza uma predição perfeita.

A Tabela 4 ilustra os resultados obtidos pelos preditores em cada grupo dos datasets sintéticos. Diante disto, para as abordagens MFMD e DMMA o valor apresentado é referente a média das médias (30 execuções $\times$ 50 datasets) enquanto que para o MEME e *Gibbs Motif Sampler* o valor é pertinente a média das melhores execuções de cada abordagem. A Tabela 5, Tabela 6 e Tabela 7 ilustram os resultados obtidos pelos preditores nos datsets semi-sintéticos, ChIP-seq e reais, respectivamente.

É importante notar na Tabela 5 e Tabela 7 que em alguns datasets, as abordagens MEME e *Gibbs Motif Sampler* obtiveram valor **0** em *precision*, *recall* e *f-score*. Em particular, na Tabela 5, o MEME alcançou este valor no dataset ABD-B e o *Gibbs Motif Sampler* nos datasets ABD-B e CHE-1. Na Tabela 7 apenas o MEME obteve valor **0**, sendo que ele foi encontrado nos datasets CREB e NFKB.

Neste contexto, fica evidente que o desvio medido pelas posições iniciais preditas por essas duas abordagens foi superior a 2 (limite definido no inicio deste capítulo, que qualifica uma posição como correta) levando a contagem de verdadeiros positivos (TP) à **0**. Por consequência, isso conduziu os valores de *precision*, *recall* e *f-score* também a **0**.

6.2.1 Análise por ranking

Os resultados foram comparados utilizando o método de dominância proposto por L. I. Kuncheva e J. J. Rodríguez (130). Neste sistema, cada abordagem recebe uma pontuação quando comparada às outras abordagens. A hierarquia de dominância é determinada pela classificação das metodologias de acordo com um escore calculado através das perdas e vitórias que cada abordagem conquistou em cada medida de desempenho. Isso corresponde ao número total de vezes que, por exemplo, a abordagem “A” conseguiu ser melhor que a abordagem “B” menos o número total de vezes que a abordagem “B” foi melhor que a abordagem “A”.

Neste contexto, as vitórias e derrotas foram definidas nos termos dos valores de *f-score* que cada estratégia foi capaz de alcançar. Uma vez que o *f-score* representa a média harmônica entre *precision* e *recall*, a magnitude do seu valor é diretamente influenciada por ambas as medidas, i.e., um valor baixo de *precision* implicará em um *f-score* baixo mesmo que o *recall* seja alto. A relação inversa também é verdadeira.

Tabela 4 – Resultados alcançados pelos preditores nos datasets sintéticos.

Grupo	Preditor	*Precision*	*Recall*	*F-Score*
1	MFMD	0.995 ± 0.007	0.998 ± 0.005	0.996 ± 0.009
	DMMA	0.970 ± 0.013	0.990 ± 0.011	0.980 ± 0.015
	MEME	0.993 ± 0.007	0.990 ± 0.009	0.991 ± 0.013
	GIBBS	0.978 ± 0.007	0.958 ± 0.009	0.967 ± 0.011
2	MFMD	0.856 ± 0.041	0.866 ± 0.043	0.860 ± 0.039
	DMMA	0.853 ± 0.039	0.859 ± 0.037	0.855 ± 0.043
	MEME	0.839 ± 0.033	0.833 ± 0.039	0.835 ± 0.035
	GIBBS	0.795 ± 0.047	0.796 ± 0.041	0.795 ± 0.037
3	MFMD	0.961 ± 0.019	0.968 ± 0.029	0.964 ± 0.011
	DMMA	0.958 ± 0.031	0.951 ± 0.027	0.954 ± 0.021
	MEME	0.931 ± 0.023	0.937 ± 0.017	0.933 ± 0.029
	GIBBS	0.769 ± 0.039	0.761 ± 0.053	0.764 ± 0.041
4	MFMD	0.995 ± 0.011	0.999 ± 0.015	0.996 ± 0.017
	DMMA	0.991 ± 0.009	0.998 ± 0.007	0.994 ± 0.017
	MEME	0.991 ± 0.011	0.996 ± 0.015	0.993 ± 0.021
	GIBBS	0.975 ± 0.033	0.980 ± 0.017	0.977 ± 0.025
5	MFMD	0.809 ± 0.077	0.815 ± 0.081	0.811 ± 0.099
	DMMA	0.811 ± 0.071	0.817 ± 0.069	0.813 ± 0.051
	MEME	0.787 ± 0.065	0.781 ± 0.081	0.783 ± 0.073
	GIBBS	0.751 ± 0.079	0.757 ± 0.067	0.753 ± 0.089
6	MFMD	0.969 ± 0.033	0.964 ± 0.051	0.966 ± 0.027
	DMMA	0.946 ± 0.147	0.940 ± 0.153	0.942 ± 0.149
	MEME	0.918 ± 0.047	0.921 ± 0.051	0.919 ± 0.033
	GIBBS	0.722 ± 0.105	0.728 ± 0.107	0.724 ± 0.115

Como pode ser observado na Tabela 8 e Tabela 9, o MFMD apresentou uma pontuação (*ranking*) superior em relação as demais abordagens comparadas para todos os datasets analisados. Diante disto, podemos observar que os ganhos no desempenho da abordagem MFMD foram referentes as modificações realizadas no procedimento da geração da população inicial do algoritmo. A boa relação entre *precision* e *recall* evidenciou que o MFMD conseguiu um equilíbrio entre os verdadeiros positivos e os falsos positivos preditos. A partir destas modificações, foi possível obter um melhor desempenho do algoritmo (melhores pontuações) em relação às demais abordagens.

O nível de significância utilizado na etapa de Correspondência de Padrão foi de 0.0001. O acréscimo deste valor acarretaria maior permissibilidade ao método trazendo como consequência um aumento no número de falsos positivos preditos. Em contrapartida, o seu decréscimo deixaria a abordagem mais "rígida" e por conseguinte um menor número de verdadeiros positivos seriam observados. Portanto, o correto ajuste deste parâmetro implica diretamente na qualidade de predição do algoritmo.

Tabela 5 – Resultados alcançados pelos preditores nos datasets semi-sintéticos.

Dataset	Preditor	*Precision*	*Recall*	*F-Score*
ABD-B	MFMD	0.475 ± 0.023	0.477 ± 0.040	0.476 ± 0.030
	DMMA	0.524 ± 0.028	0.522 ± 0.041	0.523 ± 0.026
	MEME	**0**	**0**	**0**
	GIBBS	**0**	**0**	**0**
ARNT	MFMD	0.790 ± 0.018	0.810 ± 0.025	0.800 ± 0.017
	DMMA	0.832 ± 0.024	0.770 ± 0.024	0.800 ± 0.015
	MEME	0.800	0.800	0.800
	GIBBS	0.850	0.850	0.850
BZIP910	MFMD	0.970 ± 0.011	0.972 ± 0.014	0.971 ± 0.009
	DMMA	0.977 ± 0.032	0.965 ± 0.022	0.971 ± 0.020
	MEME	0.942	0.942	0.942
	GIBBS	0.714	0.714	0.714
CG11294	MFMD	0.934 ± 0.028	0.932 ± 0.034	0.933 ± 0.021
	DMMA	0.863 ± 0.066	0.869 ± 0.054	0.866 ± 0.041
	MEME	0.866	0.866	0.866
	GIBBS	0.666	0.666	0.666
CHE–1	MFMD	0.816 ± 0.021	0.812 ± 0.030	0.814 ± 0.021
	DMMA	0.745 ± 0.011	0.735 ± 0.015	0.740 ± 0.021
	MEME	0.703	0.706	0.704
	GIBBS	**0**	**0**	**0**
LHX3	MFMD	0.999 ± 0.037	0.999 ± 0.023	0.999 ± 0.024
	DMMA	0.999 ± 0.028	0.999 ± 0.036	0.999 ± 0.023
	MEME	0.900	0.900	0.900
	GIBBS	0.750	0.750	0.750
PAX6	MFMD	0.858 ± 0.012	0.862 ± 0.011	0.860 ± 0.014
	DMMA	0.862 ± 0.017	0.858 ± 0.032	0.860 ± 0.021
	MEME	0.860	0.860	0.860
	GIBBS	0.860	0.860	0.860
RELA	MFMD	0.999 ± 0.011	0.999 ± 0.028	0.999 ± 0.032
	DMMA	0.999 ± 0.035	0.999 ± 0.063	0.999 ± 0.040
	MEME	0.888	0.888	0.888
	GIBBS	0.944	0.944	0.944
SNA	MFMD	0.849 ± 0.028	0.851 ± 0.013	0.850 ± 0.016
	DMMA	0.773 ± 0.029	0.777 ± 0.019	0.775 ± 0.018
	MEME	0.750	0.750	0.750
	GIBBS	0.850	0.850	0.850
TLX1	MFMD	0.999 ± 0.038	0.999 ± 0.032	0.999 ± 0.022
	DMMA	0.999 ± 0.024	0.999 ± 0.032	0.999 ± 0.021
	MEME	0.999	0.999	0.999
	GIBBS	0.999	0.999	0.999
USP	MFMD	0.945 ± 0.027	0.949 ± 0.013	0.947 ± 0.015
	DMMA	0.898 ± 0.038	0.890 ± 0.022	0.894 ± 0.023
	MEME	0.921	0.921	0.921
	GIBBS	0.921	0.921	0.921

Embora todos os programas comparados neste trabalho sejam baseados em modelos probabilísticos, existem diferenças consideráveis nos resultados obtidos devido ao tamanho do espaço de busca e à existência de um grande número de soluções possíveis. Algoritmos de otimização, como o MEME por exemplo, podem otimizar localmente os modelos estatísticos, no entanto, a multimodalidade inerente do espaço de busca, em geral, não permite que procedimentos guiados puramente pela otimização local explorarem muitas soluções diferentes. A arquitetura do MFMD permite uma maior flexibilidade de movimentos ao redor do espaço de busca pois aplica um processo evolutivo a uma população de possíveis

Tabela 6 – Resultados alcançados pelos preditores nos datasets ChIP–seq.

Dataset	Preditor	*Precision*	*Recall*	*F-Score*
GATA2	MFMD	0.968 ± 0.011	0.972 ± 0.021	0.970 ± 0.057
	DMMA	0.549 ± 0.014	0.541 ± 0.012	0.545 ± 0.013
	MEME	0.948	0.948	0.948
	GIBBS	0.826	0.188	0.307
GATA3	MFMD	0.971 ± 0.015	0.965 ± 0.011	0.968 ± 0.019
	DMMA	0.451 ± 0.012	0.457 ± 0.092	0.454 ± 0.022
	MEME	0.965	0.965	0.965
	GIBBS	0.440	0.094	0.156
IRF1	MFMD	0.829 ± 0.018	0.835 ± 0.023	0.832 ± 0.022
	DMMA	0.561 ± 0.026	0.567 ± 0.058	0.564 ± 0.011
	MEME	0.903	0.903	0.903
	GIBBS	0.695	0.510	0.588
NFE2L2	MFMD	0.879 ± 0.011	0.881 ± 0.031	0.880 ± 0.041
	DMMA	0.678 ± 0.005	0.682 ± 0.011	0.680 ± 0.071
	MEME	0.866	0.866	0.866
	GIBBS	0.754	0.754	0.754
TFAP2A	MFMD	0.951 ± 0.013	0.949 ± 0.070	0.950 ± 0.010
	DMMA	0.403 ± 0.024	0.401 ± 0.021	0.402 ± 0.028
	MEME	0.515	0.515	0.515
	GIBBS	0.950	0.186	0.311

soluções candidatas.

Na Tabela 10 todas as abordagens são ordenadas de acordo com o desempenho obtido na Tabela 8 e Tabela 9. Neste caso, o algoritmo posicionado mais a esquerda indica um melhor desempenho em relação ao algoritmo mais a direita (ordenação do melhor para o pior).

A Figura 20, Figura 21 e Figura 22 ilustram respectivamente os logos dos datasets semi-sintéticos, ChIP–seq e reais geradas a partir dos *motivos* encontrados pelo MFMD.

6.2.2 Análise por teste de hipótese

Esta análise consistiu na verificação e comparação dos resultados obtidos pelo MFMD com os resultados alcançados pelo DMMA utilizando métodos estatísticos de teste de hipótese. A finalidade foi apurar se existe diferença significativa entre eles e determinar qual foi a abordagem evolutiva que apresentou melhor desempenho. Foram realizados testes de significância estatística entre as diferenças dos *f-scores* alcançados por ambas abordagens.

Tabela 7 – Resultados alcançados pelos preditores nos datasets reais.

Dataset	Preditor	*Precision*	*Recall*	*F-Score*
CREB	MFMD	0.647 ± 0.024	0.578 ± 0.044	0.611 ± 0.031
	DMMA	0.764 ± 0.032	0.684 ± 0.058	0.722 ± 0.041
	MEME	**0**	**0**	**0**
	GIBBS	0.529	0.473	0.500
CRP	MFMD	0.909 ± 0.039	0.833 ± 0.033	0.869 ± 0.027
	DMMA	0.913 ± 0.037	0.875 ± 0.062	0.893 ± 0.040
	MEME	0.904	0.791	0.844
	GIBBS	0.941	0.666	0.780
HNF1	MFMD	0.772 ± 0.013	0.629 ± 0.032	0.693 ± 0.019
	DMMA	0.045 ± 0.027	0.037 ± 0.045	0.040 ± 0.031
	MEME	0.136	0.111	0.122
	GIBBS	0.500	0.222	0.307
MCB	MFMD	0.999 ± 0.030	0.667 ± 0.042	0.800 ± 0.030
	DMMA	0.900 ± 0.051	0.750 ± 0.089	0.818 ± 0.053
	MEME	0.692	0.750	0.719
	GIBBS	0.750	0.750	0.750
MEF2	MFMD	0.700 ± 0.033	0.823 ± 0.030	0.756 ± 0.024
	DMMA	0.819 ± 0.032	0.827 ± 0.066	0.823 ± 0.039
	MEME	0.705	0.705	0.705
	GIBBS	0.176	0.176	0.176
MYOD	MFMD	0.363 ± 0.016	0.380 ± 0.024	0.372 ± 0.018
	DMMA	0.336 ± 0.034	0.330 ± 0.065	0.333 ± 0.044
	MEME	0.235	0.190	0.210
	GIBBS	0.208	0.238	0.222
NFKB	MFMD	0.667 ± 0.040	0.500 ± 0.099	0.571 ± 0.062
	DMMA	0.667 ± 0.065	0.500 ± 0.112	0.571 ± 0.078
	MEME	**0**	**0**	**0**
	GIBBS	0.667	0.500	0.571
PDR3	MFMD	0.850 ± 0.035	0.944 ± 0.046	0.894 ± 0.034
	DMMA	0.894 ± 0.040	0.944 ± 0.059	0.918 ± 0.039
	MEME	0.653	0.944	0.772
	GIBBS	0.928	0.722	0.812
REB1	MFMD	0.800 ± 0.027	0.600 ± 0.025	0.685 ± 0.021
	DMMA	0.800 ± 0.049	0.600 ± 0.071	0.685 ± 0.058
	MEME	0.333	0.350	0.341
	GIBBS	0.266	0.200	0.228
SRF	MFMD	0.477 ± 0.007	0.583 ± 0.014	0.525 ± 0.008
	DMMA	0.500 ± 0.013	0.361 ± 0.022	0.419 ± 0.014
	MEME	0.440	0.611	0.511
	GIBBS	0.514	0.500	0.507
TBP	MFMD	0.657 ± 0.004	0.768 ± 0.008	0.708 ± 0.006
	DMMA	0.549 ± 0.011	0.545 ± 0.015	0.547 ± 0.012
	MEME	0.578	0.578	0.578
	GIBBS	0.308	0.347	0.326

As hipóteses a serem testadas foram:

$$\begin{cases} H_0: & \text{As amostras das abordagens são extraídas} \\ & \text{de distribuições com o mesmo valor médio.} \\ H_1: & \text{As amostras das abordagens são extraídas} \\ & \text{de distribuições diferentes.} \end{cases} \tag{6.1}$$

Tabela 8 – Vitórias e derrotas nos datasets sintéticos organizadas por grupo.

Preditor	Grupo	Vitórias	Derrotas	Total
MFMD	1	75	18	57
	2	96	37	59
	3	118	22	96
	4	31	12	19
	5	78	54	24
	6	101	18	83
DMMA	1	46	51	–5
	2	98	38	60
	3	100	37	63
	4	24	13	11
	5	88	48	40
	6	99	23	76
MEME	1	68	26	42
	2	62	73	–11
	3	67	76	–9
	4	26	7	19
	5	63	55	–2
	6	58	71	–13
GIBBS	1	17	111	–94
	2	18	126	–108
	3	0	150	–150
	4	7	56	–49
	5	37	99	–62
	6	1	147	–146

Um segundo teste foi conduzido com objetivo de comparar os resultados obtidos pelo MFMD e a melhor abordagem a partir da comparação entre os algoritmos MEME e GIBBS, baseado nos resultados apresentados na Tabela 8 e Tabela 9. Diante disto, podemos observar um melhor desempenho do algoritmo MEME em relação ao algoritmo GIBBS. Desta forma, a Tabela 11 ilustra os resultados obtidos nos testes entre o MFMD vs DMMA e MFMD vs MEME.

As análises consistiram nos seguintes passos:

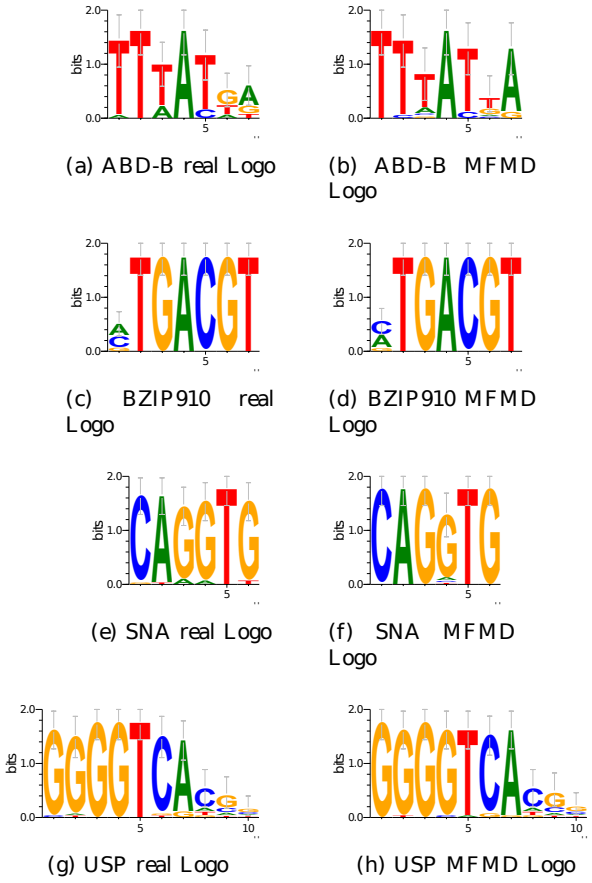
Tabela 9 – Vitórias e derrotas nos datasets semi-sintéticos, ChIP–seq e reais.

Preditor	Dataset	Vitórias	Derrotas	Total
MFMD	Semi-sintético	19	2	17
	ChIP–seq	14	1	13
	Real	25	5	20
DMMA	Semi-sintético	13	8	5
	ChIP–seq	3	12	-9
	Real	21	9	12
MEME	Semi-sintético	5	17	–12
	ChIP–seq	11	4	7
	Real	8	25	–17
GIBBS	Semi-sintético	7	17	–10
	ChIP–seq	2	13	–11
	Real	8	23	–15

Tabela 10 – Ranking dos algoritmos de acordo com as medidas de desempenho (do melhor para o pior).

Datasets sintéticos:			
MFMD	DMMA	MEME	GIBBS
Datasets semi-sintéticos:			
MFMD	DMMA	GIBBS	MEME
Datasets ChIP–seq:			
MFMD	MEME	DMMA	GIBBS
Datasets reais:			
MFMD	DMMA	GIBBS	MEME

- seleção de amostras: foram selecionados alguns conjuntos de dados para compor o teste estatístico. Foram 2 de cada grupo sintético, 2 semi-sintéticos, 5 ChIP-seq e 2 reais, totalizando 21 datasets;
- análise estatística: o parâmetro analisado foi o f-score calculado a partir das 30 execuções realizadas em cada dataset por cada algoritmo;
- o teste de Shapiro–Wilk (131) foi aplicado a cada conjunto de parâmetros. No caso da normalidade ser verificada, um teste T de Student (132) pareado foi aplicado. Caso contrário, o teste não paramétrico empregado foi o Wilcoxon (133) pareado;
- significância: o nível de significância utilizado foi de 0.05 ou 95%.



(a) ABD-B real Logo (b) ABD-B MFMD Logo

(c) BZIP910 real Logo (d) BZIP910 MFMD Logo

(e) SNA real Logo (f) SNA MFMD Logo

(g) USP real Logo (h) USP MFMD Logo

Figura 20 – Comparação entre as logos reais e as logos encontradas pelo MFMD nos datasets semi-sintéticos.

É interessante notar através da análise da Tabela 10 que o MEME obteve seu melhor desempenho nos datasets ChIP-seq, figurando em segundo lugar. Isso ficou ainda mais evidente nos dados apresentados na Tabela 6 e Tabela 11, onde fica destacado o bom comportamento deste algoritmo nos datasets GATA3 e IRF1 em relação aos demais.

O modelo probabilístico utilizado pelo MEME melhora na medida em que as amostras aumentam. Neste caso, a estimação dos parâmetros deste algoritmo ficou mais precisa devido a grande quantidade de amostras que os datasets ChIP-seq possuem. No entanto, o MFMD também se beneficia de conjuntos de dados grandes. Nos datsets ChIP-seq, embora o MEME tenha conseguido um bom desempenho, o MFMD ainda figurou em primeiro. Isso deixa claro que a principal diferença entre MEME e MFMD está concentrada no tipo a na forma como a busca global e a as heurísticas são aplicadas ao problema, já que o modelo probabilístico que ambos utilizam é muito parecido.

Isso fica ainda mais visível em datasets menores, como mostra os dados da Tabela 7, onde o MFMD obteve um desempenho consideravelmente superior ao MEME. Neste contexto, com menor número de amostras, tanto MEME quanto MFMD estimam com

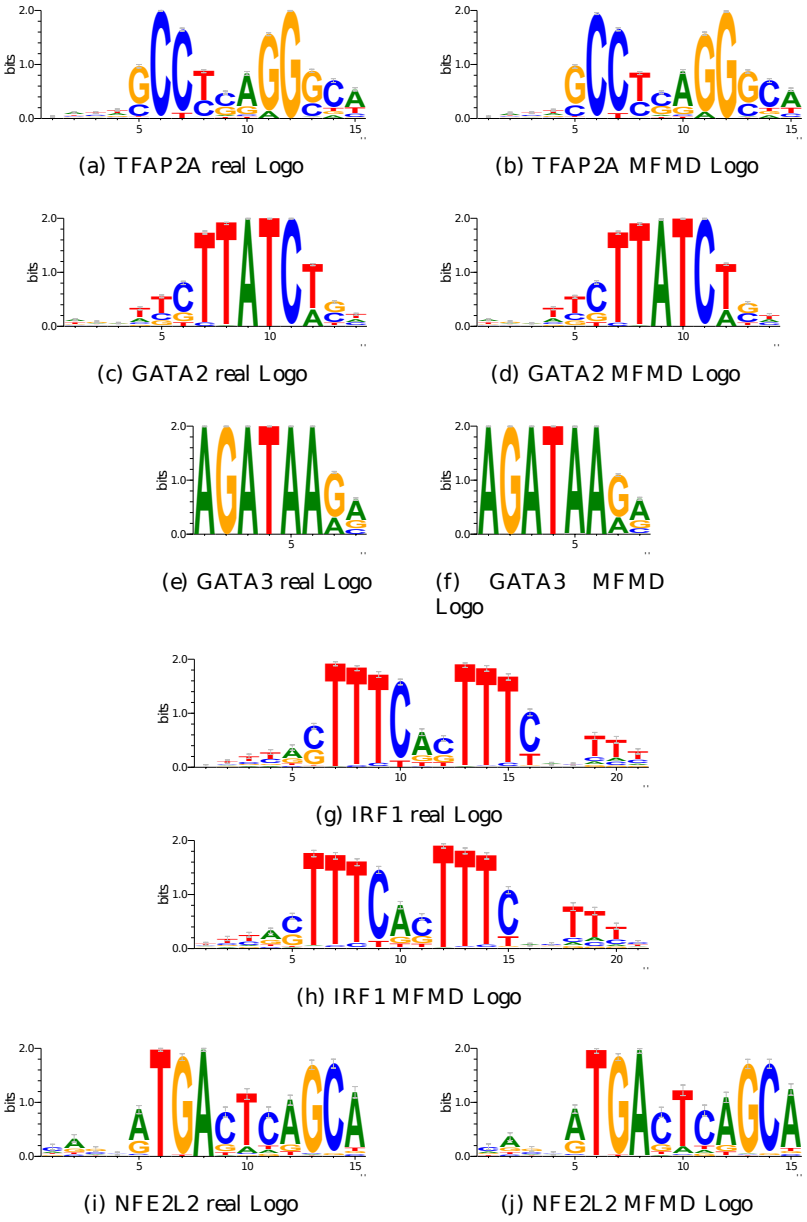
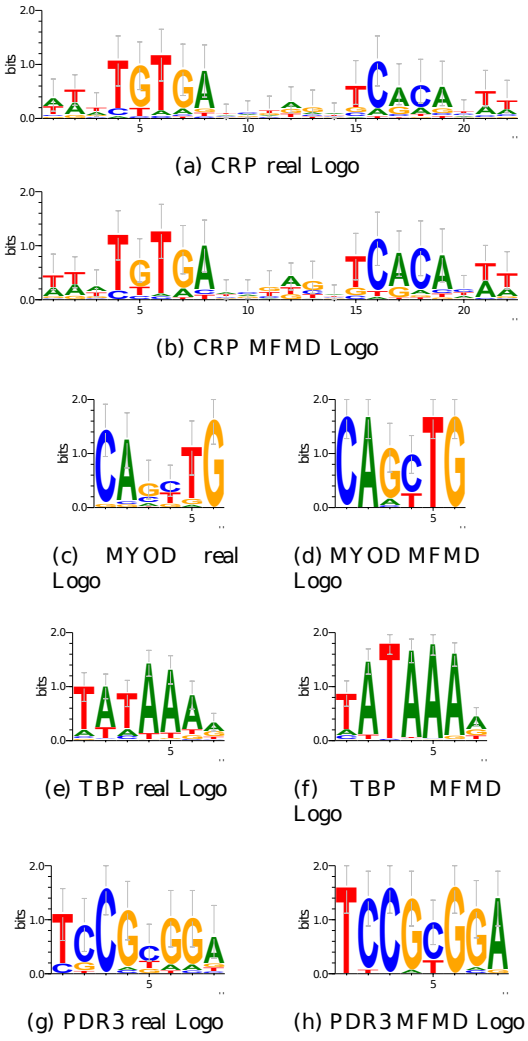


Figura 21 – Comparação entre as logos reais e as logos encontradas pelo MFMD nos datasets ChIP–seq.

menor precisão o modelo probabilístico, mas, ainda assim, o MFMD foi capaz de reconhecer um número maior de *motivos*. Em geral, o melhor desempenho alcançado pelo MFMD pode ser atribuído à sua arquitetura de otimização e a maneira mais eficaz que suas heurísticas são aplicadas, permitindo explorar com maior eficiência o espaço de busca e desta forma conseguindo alcançar melhores resultados.

Tabela 11 – Teste estatístico entre as abordagens MFMD vs DMMA e MFMD vs MEME. + Há diferença estatística (MFMD melhor); = Não há diferença entre as abordagens; - Há diferença estatística (MFMD pior).

Tipo	Grupo / Dataset	Abordagem	P-Valor	Resultado	Abordagem	P-Valor	Resultado
Sintético	Grupo 1/24	MFMD DMMA	$3.144e-05$	+	MFMD MEME	$3.577e-4$	+
	Grupo 1/40	MFMD DMMA	$3.409e-03$	+	MFMD MEME	$1.350e-11$	-
	Grupo 2/23	MFMD DMMA	$5.672e-13$	+	MFMD MEME	$2.200e-16$	+
	Grupo 2/38	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$4.304e-12$	+
	Grupo 3/21	MDMF DMMA	$3.855e-08$	+	MFMD MEME	$2.670e-05$	+
	Grupo 3/46	MFMD DMMA	$4.340e-06$	+	MFMD MEME	$2.200e-16$	+
	Grupo 4/18	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+
	Grupo 4/43	MFMD DMMA	$2.154e-13$	+	MFMD MEME	$2.004e-12$	-
	Grupo 5/12	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+
	Grupo 5/38	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+
	Grupo 6/33	MFMD DMMA	$4.770e-08$	+	MFMD MEME	$2.200e-16$	+
	Grupo 6/25	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+
Semi-sintético	CG11294	MFMD DMMA	$5.539e-11$	+	MFMD MEME	$2.200e-16$	+
	USP	MFMD DMMA	$6.078e-08$	+	MFMD MEME	$2.200e-16$	+
ChIP	GATA2	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$1.327e-3$	+
	GATA3	MFMD DMMA	$2.200e-16$	+	MFMD MEME	0.1599	=
	IRF1	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	-
	NFE2L2	MFMD DMMA	$2.200e-16$	+	MFMD MEME	0.0476	+
	TFAP2A	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+
real	SRF	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$1.401e-10$	+
	TBP	MFMD DMMA	$2.200e-16$	+	MFMD MEME	$2.200e-16$	+



(a) CRP real Logo

(b) CRP MFMD Logo

(c) MYOD real Logo

(d) MYOD MFMD Logo

(e) TBP real Logo

(f) TBP MFMD Logo

(g) PDR3 real Logo

(h) PDR3 MFMD Logo

Figura 22 – Comparação entre as logos reais e as logos encontradas pelo MFMD nos datasets reais.

7 CONSIDERAÇÕES FINAIS

Neste trabalho foi proposto um novo algoritmo para descoberta de *motivos* em sequências de DNA utilizando busca local e algoritmos evolutivos como estratégia de otimização.

A abordagem proposta, chamada MFMD, começa a partir de uma população de *motivos* gerados de forma gradativa e realiza uma pesquisa extensiva por meio de operações como recombinação, mutação e busca local.

Para demonstrar a eficiência do MFMD foram realizados vários experimentos concentrados em 4 grupos de datasets: sintéticos (datasets e *motivos* gerados algoritmicamente); semi-sintéticos (datasets gerados algoritmicamente e *motivos* extraídos de sítios reais); ChIP–seq (datasets e *motivos* extraídos de experimentos de ChIP–seq) e reais. Através das comparações realizadas entre o MFMD e outras abordagens encontradas na literatura, pode-se concluir que o MFMD foi capaz de alcançar resultados melhores na maioria do experimentos em todos os datasets.

Embora existam diversos modelos probabilísticas mais robustos que PSSM, como por exemplo *Dinucleotide Weight Matrices* (DWM) (134) e *Transcription Factor Flexible Models* (TFFM) (135), o objetivo deste trabalho foi ressaltar a eficiência da abordagem evolutiva híbrida em relação as abordagens clássicas da literatura.

Em trabalhos futuros, pretende-se investigar outras formas de representação. Apesar de existir um considerável esforço na comunidade cientifica, continua a ser um desafio complexo para os biólogos computacionais prever de forma convincente elementos regulatórios em sequências de DNA.

Os paradigmas atuais de descoberta de *motivos* podem ser vistos como uma aproximação da realidade biológica, embora esforços recentes tenham procurado incluir correlação entre posições de *motivo* (136), informações filogenéticas (137) e relações sinérgicas entre fatores de transcrição (138). À medida que a complexidade desses modelos aumenta, surge a necessidade da elaboração de algoritmos cada vez mais sofisticados que sejam capazes de encontrar soluções ótimas para esses modelos e isso se tornará cada vez mais importante ao longo do tempo.

REFERÊNCIAS

1 ALBERTS, B. et al. *Molecular biology of the cell*. 5th. ed. USA: Garland Science, 2007. Citado 8 vezes nas páginas 9, 18, 24, 25, 27, 29, 32 e 34.

2 THE Amazing Biology. Disponível em: <http://theamazingbiology.weebly.com/provas/category/all/>. Citado 2 vezes nas páginas 9 e 28.

3 CREATION Wiki. Disponível em: <http://creationwiki.org/>. Citado 2 vezes nas páginas 9 e 30.

4 CONSORTIUM, U. et al. The universal protein resource (uniprot). *Nucleic acids research*, Oxford Univ Press, v. 36, n. suppl 1, p. D190–D195, 2008. Citado 2 vezes nas páginas 9 e 33.

5 VERLI, H. *Bioinformática: da Biologia à Flexibilidade Molecular*. 1st. ed. BRA: Sociedade Brasileira de Bioquímica Molecular - SBBq, 2014. Citado 6 vezes nas páginas 10, 19, 24, 26, 32 e 34.

6 SCHNEIDER, T. D.; STEPHENS, R. M. Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, Oxford Univ Press, v. 18, n. 20, p. 6097–6100, 1990. Citado 2 vezes nas páginas 10 e 37.

7 COELLO, C. A. C.; VELDHUIZEN, D. A. V.; LAMONT, G. B. *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2002. v. 242. Citado 4 vezes nas páginas 10, 57, 58 e 59.

8 STORMO, G. D. Dna binding sites: representation and discovery. *Bioinformatics*, Oxford Univ Press, v. 16, n. 1, p. 16–23, 2000. Citado 3 vezes nas páginas 18, 35 e 71.

9 WRAY, G. A. et al. The evolution of transcriptional regulation in eukaryotes. *Molecular biology and evolution*, SMBE, v. 20, n. 9, p. 1377–1419, 2003. Citado na página 18.

10 HERTZ, G. Z.; STORMO, G. D. Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, Oxford Univ Press, v. 15, n. 7, p. 563–577, 1999. Citado 3 vezes nas páginas 18, 19 e 42.

11 STORMO, G. D.; HARTZELL, G. W. Identifying protein-binding sites from unaligned dna fragments. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 86, n. 4, p. 1183–1187, 1989. Citado 6 vezes nas páginas 18, 22, 42, 43, 65 e 80.

12 GALAS, D. J.; SCHMITZ, A. Dnaase footprinting a simple method for the detection of protein-dna binding specificity. *Nucleic acids research*, Oxford Univ Press, v. 5, n. 9, p. 3157–3170, 1978. Citado na página 18.

13 GARNER, M. M.; REVZIN, A. A gel electrophoresis method for quantifying the binding of proteins to specific dna regions: application to components of the escherichia coli lactose operon regulatory system. *Nucleic acids research*, Oxford Univ Press, v. 9, n. 13, p. 3047–3060, 1981. Citado na página 18.

14 CHAN, T.-M.; LEUNG, K.-S.; LEE, K.-H. Tfbs identification based on genetic algorithm with combined representations and adaptive post-processing. *Bioinformatics*, Oxford Univ Press, v. 24, n. 3, p. 341–349, 2008. Citado 2 vezes nas páginas 19 e 43.

15 D'HAESELEER, P. What are dna sequence motifs? *Nature biotechnology*, Nature Publishing Group, v. 24, n. 4, p. 423–425, 2006. Citado 6 vezes nas páginas 19, 34, 35, 36, 39 e 72.

16 DAS, M. K.; DAI, H.-K. A survey of dna motif finding algorithms. *BMC bioinformatics*, BioMed Central Ltd, v. 8, n. Suppl 7, p. S21, 2007. Citado 4 vezes nas páginas 19, 35, 41 e 42.

17 D'HAESELEER, P. How does dna sequence motif discovery work? *Nature biotechnology*, Nature Publishing Group, v. 24, n. 8, p. 959–961, 2006. Citado 4 vezes nas páginas 19, 40, 41 e 67.

18 DURBIN, R. et al. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. 17th. ed. UK: Cambridge University Press, 1998. Citado 3 vezes nas páginas 19, 36 e 69.

19 BAILEY, T. L. et al. Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic acids research*, Oxford Univ Press, v. 34, n. suppl 2, p. W369–W373, 2006. Citado 2 vezes nas páginas 19 e 77.

20 NEUWALD, A. F.; LIU, J. S.; LAWRENCE, C. E. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein science*, Wiley Online Library, v. 4, n. 8, p. 1618–1632, 1995. Citado na página 19.

21 TOMPA, M. et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature biotechnology*, Nature Publishing Group, v. 23, n. 1, p. 137–144, 2005. Citado na página 19.

22 SAGOT, M.-F. Spelling approximate repeated or common motifs using a suffix tree. In: *LATIN'98: Theoretical Informatics*. [S.l.]: Springer, 1998. p. 374–390. Citado na página 19.

23 PAVESI, G.; MAURI, G.; PESOLE, G. An algorithm for finding signals of unknown length in dna sequences. *Bioinformatics*, Oxford Univ Press, v. 17, n. suppl 1, p. S207–S214, 2001. Citado 2 vezes nas páginas 19 e 42.

24 KAYA, M. Mogamod: Multi-objective genetic algorithm for motif discovery. *Expert Systems with Applications*, Elsevier, v. 36, n. 2, p. 1039–1047, 2009. Citado 3 vezes nas páginas 20, 39 e 40.

25 ALON, U. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, Nature Publishing Group, v. 8, n. 6, p. 450–461, 2007. Citado 2 vezes nas páginas 20 e 36.

26 TAVARES, W. et al. Bactérias gram-positivas problemas: resistência do estafilococo, do enterococo e do pneumococo aos antimicrobianos. *Revista da Sociedade Brasileira de Medicina Tropical*, SciELO Brasil, v. 33, n. 3, p. 281–301, 2000. Citado na página 20.

27 CRUVINEL, W. d. M. et al. Immune system: Part i. fundamentals of innate immunity with emphasis on molecular and cellular mechanisms of inflammatory response. *Revista brasileira de reumatologia*, SciELO Brasil, v. 50, n. 4, p. 434–447, 2010. Citado na página 20.

28 FOGEL, G. B. et al. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. *Nucleic Acids Research*, Oxford Univ Press, v. 32, n. 13, p. 3826–3835, 2004. Citado 3 vezes nas páginas 22, 43 e 72.

29 LI, M.; MA, B.; WANG, L. Finding similar regions in many strings. In: ACM. *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. [S.l.], 1999. p. 473–482. Citado na página 22.

30 ZAHA, A.; FERREIRA, H. B.; PASSAGLIA, L. M. P. *Biologia Molecular Básica*. 3th. ed. BRA: Mercado Aberto, 2003. Citado 5 vezes nas páginas 23, 24, 26, 28 e 29.

31 NELSON, D. L.; COX, M. M. Princípios de bioquímica de lehninger. In: *Principios de Bioquimica de Lehninger*. [S.l.]: Artmed, 2014. v. 6. Citado 6 vezes nas páginas 23, 24, 26, 28, 29 e 31.

32 SNUSTAD, D. P.; SIMMONS, M. J. *Principles of Genetics*. 6th. ed. USA: Biochemical Education, 2012. Citado na página 27.

33 WAGNER, J. *Gibbs Sampling for Gapped Motif Discovery in Proteins*. Dissertação (Mestrado) — B. Sc., University of Alberta, 2005. Citado na página 31.

34 LONES, M. A.; TYRRELL, A. M. The evolutionary computation approach to motif discovery in biological sequences. In: ACM. *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*. [S.l.], 2005. p. 1–11. Citado 2 vezes nas páginas 31 e 44.

35 BORK, P.; KOONIN, E. V. Protein sequence motifs. *Current opinion in structural biology*, Elsevier, v. 6, n. 3, p. 366–376, 1996. Citado na página 31.

36 EMBL-EBI. *Protein Classification*. 2015. <http://www.ebi.ac.uk/training/online/course/introduction-protein-classification-ebi/protein-classification>. [Acessado em 24/10/2015]. Citado na página 31.

37 ACHAR, A.; SÆTROM, P. Rna motif discovery: a computational overview. *Biology direct*, Springer, v. 10, n. 1, p. 1–22, 2015. Citado na página 35.

38 DJORDJEVIC, M.; SENGUPTA, A. M.; SHRAIMAN, B. I. A biophysical approach to transcription factor binding site discovery. *Genome research*, Cold Spring Harbor Lab, v. 13, n. 11, p. 2381–2390, 2003. Citado na página 35.

39 HENDRIX, D. K.; BRENNER, S. E.; HOLBROOK, S. R. Rna structural motifs: building blocks of a modular biomolecule. *Quarterly reviews of biophysics*, Cambridge Univ Press, v. 38, n. 03, p. 221–243, 2005. Citado 2 vezes nas páginas 37 e 38.

40 MOORE, P. B. Structural motifs in rna. *Annual review of biochemistry*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 68, n. 1, p. 287–300, 1999. Citado na página 38.

41 GUTELL, R. Comparative sequence analysis and the structure of 16s and 23s rrna. *Ribosomal RNA: structure, evolution, processing, and function in protein biosynthesis*, CRC Press Boca Raton, Florida, p. 111–128, 1996. Citado na página 38.

42 BRENOWITZ, M. et al. Quantitative dnase footprint titration: a method for studying protein-dna interactions. *Methods in enzymology*, v. 130, p. 132, 1986. Citado na página 39.

43 ANDREWS, N. C.; FALLER, D. V. A rapid micropreparation technique for extraction of dna-binding proteins from limiting numbers of mammalian cells. *Nucleic acids research*, Oxford University Press, v. 19, n. 9, p. 2499, 1991. Citado na página 39.

44 ZHANG, Y. et al. Model-based analysis of chip-seq (macs). *Genome biology*, BioMed Central Ltd, v. 9, n. 9, p. R137, 2008. Citado na página 39.

45 ULMASOV, T. et al. Aux/iaa proteins repress expression of reporter genes containing natural and highly active synthetic auxin response elements. *The Plant Cell*, Am Soc Plant Biol, v. 9, n. 11, p. 1963–1971, 1997. Citado na página 39.

46 STOLTENBURG, R.; REINEMANN, C.; STREHLITZ, B. Selex—a (r) evolutionary method to generate high-affinity nucleic acid ligands. *Biomolecular engineering*, Elsevier, v. 24, n. 4, p. 381–403, 2007. Citado na página 39.

47 LIHU, A.; HOLBAN, Ş. A review of ensemble methods for de novo motif discovery in chip-seq data. *Briefings in bioinformatics*, Oxford Univ Press, p. bbv022, 2015. Citado 2 vezes nas páginas 39 e 40.

48 ZIA, A.; MOSES, A. M. Towards a theoretical understanding of false positives in dna motif finding. *BMC bioinformatics*, BioMed Central Ltd, v. 13, n. 1, p. 151, 2012. Citado 2 vezes nas páginas 41 e 72.

49 CORNISH-BOWDEN, A. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic acids research*, Oxford University Press, v. 13, n. 9, p. 3021, 1985. Citado na página 41.

50 HELDEN, J. V.; RIOS, A. F.; COLLADO-VIDES, J. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic acids research*, Oxford Univ Press, v. 28, n. 8, p. 1808–1818, 2000. Citado na página 41.

51 SINHA, S.; TOMPA, M. A statistical method for finding transcription factor binding sites. In: *ISMB*. [S.l.: s.n.], 2000. v. 8, p. 344–354. Citado na página 41.

52 LAWRENCE, C. E. et al. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *science*, American Association for the Advancement of Science, v. 262, n. 5131, p. 208–214, 1993. Citado na página 42.

53 BAILEY, T. L.; ELKAN, C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine learning*, Springer, v. 21, n. 1-2, p. 51–80, 1995. Citado na página 43.

54 CORNE, D.; MEADE, A.; SIBLY, R. Evolving core promoter signal motifs. In: IEEE. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*. [S.l.], 2001. v. 2, p. 1162–1169. Citado na página 43.

55 CHE, D.; SONG, Y.; RASHEED, K. Mdga: motif discovery using a genetic algorithm. In: ACM. *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. [S.l.], 2005. p. 447–452. Citado na página 43.

56 CONGDON, C. B. et al. Preliminary results for gami: A genetic algorithms approach to motif inference. In: IEEE. *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on*. [S.l.], 2005. p. 1–8. Citado na página 43.

57 LUO, J.-w.; WANG, T. Motif discovery using an immune genetic algorithm. *Journal of theoretical biology*, Elsevier, v. 264, n. 2, p. 319–325, 2010. Citado na página 44.

58 GOLDBARG, M. C.; GOLDBARG, E. G.; LUNA, H. P. L. *Otimização combinatória e meta-heurísticas: algoritmos e aplicações*. [S.l.]: Elsevier, 2005. v. 2. Citado 2 vezes nas páginas 45 e 53.

59 CAMPELLO, R. E. *Algoritmos e heurísticas: desenvolvimento e avaliação de performance*. [S.l.]: EDUFF, 1994. Citado na página 45.

60 MOSCATO, P. et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, v. 826, p. 1989, 1989. Citado 2 vezes nas páginas 45 e 61.

61 CUNHA, C. B. da; BONASSER, U. de O.; ABRAHÃO, F. T. M. Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante. In: *XVI Congresso da Anpet*. [S.l.: s.n.], 2002. Citado na página 46.

62 GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. [S.l.]: Elsevier, 2005. v. 2. Citado na página 46.

63 RICH, E.; KNIGHT, K. *Inteligenqa Artificial*. [S.l.]: Makron Books do Brasil, Sao Paulo, 1993. Citado na página 46.

64 ZANAKIS, S. H.; EVANS, J. R. Heuristic "optimization": Why, when, and how to use it. *Interfaces*, INFORMS, v. 11, n. 5, p. 84–91, 1981. Citado na página 46.

65 NORVIG, P.; RUSSELL, S. *Inteligência Artificial*. 3th. ed. USA: Elsevier, 2013. Citado na página 46.

66 HART, E.; ROSS, P.; NELSON, J. Solving a real-world problem using an evolving heuristically driven schedule builder. *Evolutionary Computation*, MIT Press, v. 6, n. 1, p. 61–80, 1998. Citado na página 46.

67 VERDEGAY, J. L.; YAGER, R. R.; BONISSONE, P. P. On heuristics as a fundamental constituent of soft computing. *Fuzzy sets and systems*, Elsevier, v. 159, n. 7, p. 846–855, 2008. Citado na página 47.

68 GLOVER, F. W.; KOCHENBERGER, G. A. *Handbook of metaheuristics*. [S.l.]: Springer Science & Business Media, 2006. v. 57. Citado na página 47.

69 BURKE, E. et al. An emerging direction in modern search technology. *Handbook of Metaheuristics*, v. 2, p. 457474. Citado na página 47.

70 MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado na página 47.

71 HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. *European journal of operational research*, Elsevier, v. 130, n. 3, p. 449–467, 2001. Citado na página 48.

72 HANSEN, P.; MLADENOVIĆ, N. *Variable neighborhood search*. [S.l.]: Springer, 2014. Citado na página 48.

73 FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995. Citado 2 vezes nas páginas 50 e 68.

74 FEO, T. A.; RESENDE, M. G.; SMITH, S. H. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, INFORMS, v. 42, n. 5, p. 860–878, 1994. Citado na página 50.

75 RESENDE, M. G.; RIBEIRO, C. C. Optimization by grasp. Springer, 2016. Citado 2 vezes nas páginas 51 e 52.

76 HART, J. P.; SHOGAN, A. W. Semi-greedy heuristics: An empirical study. *Operations Research Letters*, Elsevier, v. 6, n. 3, p. 107–114, 1987. Citado na página 51.

77 METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, AIP, v. 21, n. 6, p. 1087–1092, 1953. Citado na página 53.

78 KIRKPATRICK, S. et al. Optimization by simulated annealing. *science*, World Scientific, v. 220, n. 4598, p. 671–680, 1983. Citado 2 vezes nas páginas 53 e 55.

79 AARTS, E. H.; KORST, J. H. Simulated annealing. *ISSUES*, v. 1, p. 16, 1988. Citado 2 vezes nas páginas 53 e 55.

80 HWANG, C.-R. Simulated annealing: theory and applications. *Acta Applicandae Mathematicae*, Springer, v. 12, n. 1, p. 108–111, 1988. Citado na página 54.

81 LUNDY, M.; MEES, A. Convergence of an annealing algorithm. *Mathematical programming*, Springer, v. 34, n. 1, p. 111–124, 1986. Citado na página 55.

82 ZUBEN, F. J. V. Computação evolutiva: uma abordagem pragmática. *Tutorial: Notas de Aula da disciplina IA707, Faculdade de Engenharia Elétrica e de Computação-Universidade Estadual de Campinas*, 2000. Citado 2 vezes nas páginas 55 e 56.

83 LINDEN, R. *Algoritmos genéticos (2a ediçao)*. [S.l.]: Brasport, 2008. Citado na página 55.

84 JONG, K. A. D. *Evolutionary Computation: A Unified Approach*. UK: The MIT Press, 2006. Citado 4 vezes nas páginas 56, 58, 59 e 60.

85 GABRIEL, P. H. R.; DELBEM, A. C. B. *Fundamentos de algoritmos evolutivos*. [S.l.]: ICMC-USP, 2008. Citado na página 57.

86 DAWKINS, R. et al. *The selfish gene*. [S.l.]: Oxford university press, 2016. Citado na página 61.

87 NORMAN, M. G.; MOSCATO, P. A competitive and cooperative approach to complex combinatorial search. In: CITESEER. *Proceedings of the 20th Informatics and Operations Research Meeting*. [S.l.], 1991. p. 3–15. Citado na página 61.

88 HODGSON, R. Memetic algorithms and the molecular geometry optimization problem. In: IEEE. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. [S.l.], 2000. v. 1, p. 625–632. Citado na página 62.

89 MOSCATO, P.; COTTA, C.; MENDES, A. Memetic algorithms. In: *New optimization techniques in engineering*. [S.l.]: Springer, 2004. p. 53–85. Citado 2 vezes nas páginas 62 e 63.

90 RADCLIFFE, N. J. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, Springer, v. 10, n. 4, p. 339–384, 1994. Citado na página 62.

91 COLL, P. E.; DURÁN, G. A.; MOSCATO, P. On worst-case and comparative analysis as design principles for efficient recombination operators: A graph coloring case study. *New Ideas in Optimization*, McGraw-Hill, Maidenhead, Berkshire, England, UK, p. 279–294, 1999. Citado na página 62.

92 BEASLEY, J. E.; CHU, P. C. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, Elsevier, v. 94, n. 2, p. 392–404, 1996. Citado na página 63.

93 BERRETTA, R.; MOSCATO, P. The number partitioning problem: An open challenge for evolutionary computation? In: MCGRAW-HILL LTD., UK. *New ideas in optimization*. [S.l.], 1999. p. 261–278. Citado na página 63.

94 AREIBI, S. An integrated genetic algorithm with dynamic hill climbing for vlsi circuit partitioning. In: CITESEER. *GECCO 2000*. [S.l.], 2000. p. 97–102. Citado na página 63.

95 ABBASS, H. A. A memetic pareto evolutionary approach to artificial neural networks. In: SPRINGER. *Australian Joint Conference on Artificial Intelligence*. [S.l.], 2001. p. 1–12. Citado na página 64.

96 DAVIDOR, Y.; BEN-KIKI, O. The interplay among the genetic algorithm operators: Information theory tools used in a holistic way. In: *PPSN*. [S.l.: s.n.], 1992. p. 77–86. Citado na página 64.

97 HULIN, M. An optimal stop criterion for genetic algorithms: A bayesian approach. In: *ICGA*. [S.l.: s.n.], 1997. p. 135–143. Citado na página 64.

98 GARBELINI, J. C.; KASHIWABARA, A. Y.; SANCHES, D. S. Discovery motifs by evolutionary computation. In: ACM. *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. [S.l.], 2016. p. 1463–1464. Citado na página 65.

99 GARBELINI, J. M. C.; KASHIWABARA, A. Y.; SANCHES, D. S. Discovery biological motifs using heuristics approaches. In: IEEE. *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. [S.l.], 2016. p. 175–180. Citado na página 65.

100 WASSERMAN, W. W.; SANDELIN, A. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, Nature Publishing Group, v. 5, n. 4, p. 276–287, 2004. Citado na página 67.

101 TATUSOV, R.; LIPMAN, D. Dust, in the ncbi. *Toolkit available at http://blast. wustl. edu/pub/dust*. Citado na página 67.

102 SMIT, A. F.; HUBLEY, R.; GREEN, P. *RepeatMasker*. 1996. Citado na página 67.

103 ANDRONESCU, M.; RASTEGARI, B. Motif-grasp and motif-ils: Two new stochastic local search algorithms for motif finding. In: *Mini Workshop on Stochastic Search Algorithms*. [S.l.: s.n.], 2003. Citado na página 68.

104 BENOS, P. V.; BULYK, M. L.; STORMO, G. D. Additivity in protein–dna interactions: how good an approximation is it? *Nucleic acids research*, Oxford Univ Press, v. 30, n. 20, p. 4442–4451, 2002. Citado na página 70.

105 NAGARAJAN, N.; JONES, N.; KEICH, U. Computing the p-value of the information content from an alignment of multiple sequences. *Bioinformatics*, Oxford Univ Press, v. 21, n. suppl 1, p. i311–i318, 2005. Citado na página 74.

106 SCIENCE, B. S. for the Philosophy of; SCIENCE, B. S. for the History of. *The British journal for the philosophy of science*. [S.l.]: Aberdeen University Press, 1950. v. 1. Citado na página 74.

107 THOMPSON, W.; ROUCHKA, E. C.; LAWRENCE, C. E. Gibbs recursive sampler: finding transcription factor binding sites. *Nucleic acids research*, Oxford Univ Press, v. 31, n. 13, p. 3580–3585, 2003. Citado na página 77.

108 SHAW, W. M.; BURGIN, R.; HOWELL, P. Performance standards and evaluations in ir test collections: Cluster-based retrieval models. *Information Processing & Management*, Elsevier, v. 33, n. 1, p. 1–14, 1997. Citado na página 77.

109 SANDELIN, A. et al. Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic acids research*, Oxford Univ Press, v. 32, n. suppl 1, p. D91–D94, 2004. Citado na página 78.

110 BLANCO, E. et al. Abs: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic acids research*, Oxford Univ Press, v. 34, n. suppl 1, p. D63–D67, 2006. Citado na página 80.

111 ZHU, J.; ZHANG, M. Q. Scpd: a promoter database of the yeast saccharomyces cerevisiae. *Bioinformatics*, Oxford Univ Press, v. 15, n. 7, p. 607–611, 1999. Citado na página 80.

112 PURVES, G. D. et al. *Neuroscience (ed.)*. [S.l.]: Sinauer Associates, 2008. Citado na página 80.

113 DIBNER, C.; SCHIBLER, U.; ALBRECHT, U. The mammalian circadian timing system: organization and coordination of central and peripheral clocks. *Annual review of physiology*, Annual Reviews, v. 72, p. 517–549, 2010. Citado na página 80.

114 LAWSON, C. L. et al. Catabolite activator protein: Dna binding and transcription activation. *Current opinion in structural biology*, Elsevier, v. 14, n. 1, p. 10–20, 2004. Citado na página 80.

115 HERNANDEZ, L. M. R. et al. Berry phenolic compounds increase expression of hepatocyte nuclear factor-1α (hnf-1α) in caco-2 and normal colon cells due to high affinities with transcription and dimerization domains of hnf-1α. *PloS one*, Public Library of Science, v. 10, n. 9, p. e0138768, 2015. Citado na página 80.

116 POTTHOFF, M. J.; OLSON, E. N. Mef2: a central regulator of diverse developmental programs. *Development*, The Company of Biologists Ltd, v. 134, n. 23, p. 4131–4140, 2007. Citado na página 80.

117 LILLY, B. et al. D-mef2: a mads box transcription factor expressed in differentiating mesoderm and muscle cell lineages during drosophila embryogenesis. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 91, n. 12, p. 5662–5666, 1994. Citado na página 80.

118 RUDNICKI, M. A. et al. Myod or myf-5 is required for the formation of skeletal muscle. *Cell*, Elsevier, v. 75, n. 7, p. 1351–1359, 1993. Citado na página 80.

119 ALBENSI, B. C.; MATTSON, M. P. Evidence for the involvement of tnf and nf-kb in hippocampal synaptic plasticity. *Synapse-New York*, New York: Alan R. Liss, Inc., c1987-, v. 35, n. 2, p. 151–159, 2000. Citado na página 81.

120 SHORE, P.; SHARROCKS, A. D. The mads-box family of transcription factors. *European Journal of Biochemistry*, Wiley Online Library, v. 229, n. 1, p. 1–13, 1995. Citado na página 81.

121 DALTON, S. et al. Isolation and characterization of srf accessory proteins. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, The Royal Society, v. 340, n. 1293, p. 325–332, 1993. Citado na página 81.

122 LEE, T. I.; YOUNG, R. A. Transcription of eukaryotic protein-coding genes. *Annual review of genetics*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 34, n. 1, p. 77–137, 2000. Citado na página 81.

123 KORNBERG, R. D. The molecular basis of eukaryotic transcription. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 104, n. 32, p. 12955–12961, 2007. Citado na página 81.

124 MAMNUN, Y. M. et al. The yeast zinc finger regulators pdr1p and pdr3p control pleiotropic drug resistance (pdr) as homo-and heterodimers in vivo. *Molecular microbiology*, Wiley Online Library, v. 46, n. 5, p. 1429–1440, 2002. Citado na página 81.

125 MACPHERSON, S.; LAROCHELLE, M.; TURCOTTE, B. A fungal family of transcriptional regulators: the zinc cluster proteins. *Microbiology and Molecular Biology Reviews*, Am Soc Microbiol, v. 70, n. 3, p. 583–604, 2006. Citado na página 81.

126 MORROW, B. E.; JOHNSON, S. P.; WARNER, J. R. Proteins that bind to the yeast rdna enhancer. *Journal of biological chemistry*, ASBMB, v. 264, n. 15, p. 9061–9068, 1989. Citado na página 81.

127 NASMYTH, K. At the heart of the budding yeast cell cycle. *Trends in Genetics*, Elsevier, v. 12, n. 10, p. 405–412, 1996. Citado na página 81.

128 IYER, V. R. et al. Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, Nature Publishing Group, v. 409, n. 6819, p. 533–538, 2001. Citado na página 81.

129 BASERGA, R. The cell cycle. *New England Journal of Medicine*, Mass Medical Soc, v. 304, n. 8, p. 453–459, 1981. Citado na página 81.

130 KUNCHEVA, L. I.; RODRÍGUEZ, J. J. An experimental study on rotation forest ensembles. In: *Multiple Classifier Systems*. [S.l.]: Springer, 2007. p. 459–468. Citado na página 82.

131 SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, JSTOR, v. 52, n. 3/4, p. 591–611, 1965. Citado na página 88.

132 MANKIEWICZ, R. *The story of mathematics*. [S.l.]: Cassell, 2000. Citado na página 88.

133 WILCOXON, F. Individual comparisons by ranking methods. *Biometrics bulletin*, JSTOR, v. 1, n. 6, p. 80–83, 1945. Citado na página 88.

134 SIDDHARTHAN, R. Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PloS one*, Public Library of Science, v. 5, n. 3, p. e9722, 2010. Citado na página 93.

135 MATHELIER, A.; WASSERMAN, W. W. The next generation of transcription factor binding site prediction. *PLoS Comput Biol*, Public Library of Science, v. 9, n. 9, p. e1003214, 2013. Citado na página 93.

136 ZHOU, Q.; LIU, J. S. Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics*, Oxford Univ Press, v. 20, n. 6, p. 909–916, 2004. Citado na página 93.

137 SIDDHARTHAN, R.; SIGGIA, E. D.; NIMWEGEN, E. V. Phylogibbs: a gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol*, v. 1, n. 7, p. e67, 2005. Citado na página 93.

138 GUPTA, M.; LIU, J. S. De novo cis-regulatory module elicitation for eukaryotic genomes. *Proceedings of the National Academy of Sciences of the United States of America*, National Acad Sciences, v. 102, n. 20, p. 7079–7084, 2005. Citado na página 93.