UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
ESCOLA DE ENGENHARIA DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA BIOMÉDICA

RODRIGO GOMES DE SOUZA

ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS UTILIZANDO OTIMIZAÇÃO DIALÉTICA

Recife – PE
2014

RODRIGO GOMES DE SOUZA

ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS UTILIZANDO OTIMIZAÇÃO DIALÉTICA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Biomédica, da Universidade Federal de Pernambuco, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Biomédica.

ORIENTADOR: Wellington Pinheiro dos Santos

CO-ORIENTADOR: Ricardo Yara

Recife – PE

2014

Catalogação na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

S729a Souza, Rodrigo Gomes de.
Alinhamento múltiplo de seqüências utilizando otimização dialética / Rodrigo Gomes de Souza. - Recife: O Autor, 2014.
89 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Wellington Pinheiro dos Santos.
Coorientador: Prof. Dr. Ricardo Yara.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Biomédica, 2014.
Inclui Referências e Apêndice.

1. Engenharia Biomédica. 2. Alinhamento múltiplo. 3. Dialética. 4. Otimização. I. Santos, Wellington Pinheiro dos. (Orientador). II. Yara, Ricardo. (Coorientador). III. Título.

UFPE

610.28 CDD (22. ed.) BCTG/2014-160

RODRIGO GOMES DE SOUZA

ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS UTILIZANDO OTIMIZAÇÃO DIALÉTICA

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Biomédica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: Prof. Wellington Pinheiro dos Santos, Doutor pela Universidade Federal de Campina Grande, Campina Grande – Brasil

Banca Examinadora:

Prof. Wellington Pinheiro dos Santos, UFPE
Doutor pela Universidade Federal de Campina Grande, Campina Grande – Brasil

Prof. Ricardo Emmanuel de Souza, UFPE
Doutor pela Universidade Federal de Pernambuco, Recife, Brasil

Prof. Sérgio Ricardo de Melo Queiroz, UFPE
Doutor pela Université Pierre et Marie Curie, Paris – França

Coordenador do PPGEB: _____

Prof. Dr. Rosa Amalia Fireman Dutra

Recife, Março 2014.

DEDICATÓRIA

Dedico este trabalho à minha família e especialmente à minha filha Júlia por me conceder um novo propósito e uma nova visão a tudo.

AGRADECIMENTOS

Gostaria de agradecer à minha família pois, sem ela, seria praticamente impossível a conclusão deste trabalho. Agradecer imensamente aos meus Pais, Maria e Valdemir, pela motivação e suporte dados, principalmente no que se diz respeito à educação e atenção à Júlia. Igualmente sou muito grato à minha noiva, Caroline Neves, pela fonte de carinho, motivação, inspiração, coragem e confiança que sempre representou para mim.

Agradeço ao meu orientador e amigo, Wellington dos Santos, pela imensurável ajuda através de sugestões, questionamentos e confiança emitidas durante todo o trabalho, pelo convite em participar deste tão especial programa de pós-graduação e pelos bons conselhos a respeito da vida acadêmica e profissional.

Também gostaria de agradecer aos professores deste programa de pós-graduação, pela grande oportunidade e experiência vividas além da visão crítica interdisciplinar do mundo que adquiri ao ingressar no programa. Gostaria de agradecer especialmente ao prof. Ricardo Yara pela paciência eterna com minhas diversas dúvidas biológicas. Gostaria de agradecer ao Prof. Aluízio Azevedo, do Centro de Informática, pela oportunidade de ser seu aluno. Também agradeço aos meus colegas de turma pela união e cooperação entre todos nós.

Por fim, agradeço à Júlia pela incondicional alegria que me transmite todos os dias pois com ela tudo sempre fica mais fácil e prazeroso.

RESUMO

Este trabalho propõe uma abordagem baseada no método dialético de otimização para resolver o problema do alinhamento múltiplo de sequências (MSA). Nesta abordagem, problemas de múltiplo alinhamento de sequências são vistos como problemas de otimização, onde os candidatos à solução são modelados como vetores cujas componentes representam as posições das lacunas ao longo das sequências. Além disso, os candidatos a solução são avaliados através de uma função objetivo que é sugerida como uma composição de funções para pontuação de correspondências, funções para penalização e pontuação por aspectos desejados e não-desejados. Com o objetivo de testar computacionalmente esta proposta, foram criados um conjunto sintético de dados, composto de 50 grupos de 4 sequências e um modelo equivalente baseado em algoritmos genéticos. A representação de candidatos à solução baseada em posições trouxe um problema com relação à quantidade de lacunas que deveria ser utilizada no alinhamento de cada um dos 50 grupos de sequências. Como solução, a ferramenta ClustalW foi aplicada, em cada grupo de sequências, para produzir um alinhamento múltiplo, o qual foi utilizado para fornecer informações sobre a quantidade de lacunas utilizada. Os alinhamentos realizados pelo ClustalW também foram avaliados pela função objetivo proposta, para a produção de resultados comparáveis. Os experimentos foram definidos sob três abordagens quanto ao número de lacunas utilizado. Na primeira abordagem, para o alinhamento de cada grupo de sequências foi utilizada uma quantidade fixa de lacunas e equivalente à metade do comprimento das sequências, enquanto que na segunda abordagem, foi utilizada um número de lacunas igual ao usado pelo ClustalW. Na terceira abordagem, o número de lacunas usado por cada candidato à solução existente da população inicial foi definido com um valor escolhido aleatoriamente entre os valores que correspondem a 5% e 50% do comprimento. A cada abordagem, os experimentos foram refeitos utilizando-se uma variação na qual o alinhamento produzido pelo Clustal era inserido foi população inicial, em um processo conhecido como semeadura. Todos os experimentos foram primeiramente realizados utilizando o modelo alternativo, baseado em algoritmos genéticos, a fim de validar representação e função objetivo sugeridas, e, foram refeitos em seguida utilizando o método baseado em otimização dialética. Os resultados obtidos por ambos modelos foram comparados com os resultados obtidos pelos alinhamentos produzidos pelo ClustalW através do teste não-paramétrico de Wilcoxon para amostras pareadas. Em comparação com o algoritmo ClustalW, o modelo baseado no método dialético de otimização provou ser capaz de produzir alinhamentos de altos scores como também de realizar melhorias significativas nos alinhamentos encontrados pelo ClustalW.

Palavras-chaves: alinhamento múltiplo, dialética, otimização

ABSTRACT

This work proposes an approach based on dialectical method of optimization to solve the problem of multiple sequence alignment (MSA). In this approach, problems of multiple sequence alignment are seen as optimization problems, where the solution candidates are modeled as a vector whose components represent the positions of the gaps along the sequences. Moreover, the solution candidates are evaluated through an objective function which is suggested as a composition of functions for scoring matches as well as wanted and unwanted aspects. In order to test computationally this proposal, a synthetic data set consisting of 50 groups of 4 sequences and a similar model based on genetic algorithms were created. The representation of solution candidates based on gaps positions yields a problem regarding the amount of gaps in the alignment should be used for each of the 50 groups of sequences. As a solution, the ClustalW tool was applied to each group of sequences to produce a multiple sequences alignment which has been used to provide information about the amount of gaps used. Alignments made by the ClustalW were evaluated by objective function proposed to produce comparable results. The experiments were set under three approaches according to the number of gaps used. In the first approach, it was used a fixed amount of gaps equals to fifty percent of the length of the sequences to perform alignment of each group of sequences, whereas in the second approach, it was used the same amount of gaps used by ClustalW. In the third approach, the amount of gaps used by each candidate solution existing in the initial population was set to a random value selected from a range between 5% to 50% of the length of sequences. In each approach, the experiments were remade using a variation in which the alignment produced by Clustal was inserted was initial population in a process known as seeding. All experiments were performed first using the alternative model, based on genetic algorithms, in order to validate representation and suggested objective function, and then were remade using the optimization method based on dialectics. The results obtained by both models were then compared with that obtained by the ClustalW alignments by using the non-parametric Wilcoxon test for paired samples. When compared with the ClustalW, the model based on dialectical method of optimization proved be able of producing high score alignments as well as to provide significant improvements in the alignments found by the ClustalW tool.

Keywords: multiple alignment, dialectics, optimization

ÍNDICE DE TABELAS

Tabela 2.1: Código Genético.....8
Tabela 6.1: Valores dos parâmetros utilizados no ClustalW.....61
Tabela 6.2: Exemplos de polos gerados na população inicial para cada uma das três abordagens onde λ corresponde ao comprimento das N sequências a serem alinhadas, L o total de posições de lacunas a serem inicializadas e L Clustal a quantidade de lacunas utilizadas pelo Clustal.....64
Tabela 6.3: Valores utilizados para os multiplicadores das funções componentes.....69
Tabela 6.4:Valores utilizados para os parâmetros do GA.....71
Tabela 6.5: Valores médios, máximos, mínimos e de significância obtidos pelo ODM em cada abordagem.....75
Tabela 6.6: Valores médios, máximos, mínimos e de significância obtidos pelo GA em cada abordagem.....77

ÍNDICE DE FIGURAS

Figura 2.1: Gregor Johann Mendel (1822– 1884).....5
Figura 2.2: Estruturas das moléculas de DNA (à esq.) e RNA (à dir.).....7
Figura 2.3: Esquema geral da expressão gênica na produção de proteínas.....9
Figura 2.4: Processo de Transcrição de DNA em mRNA.....10
Figura 2.5: Formas de estrutura secundária: alfa hélice, beta folha e bobina aleatória.....12
Figura 2.6: Níveis de Estruturas de Proteínas.....13
Figura 3.1: Exemplo de alinhamento múltiplo de sequências de proteínas.....17
Figura 3.2: Exemplo de dendrograma de similaridade.....21
Figura 3.3: Deficiência do método progressivo em não reavaliar lacunas inseridas inicialmente 22
Figura 3.4: Fluxograma geral do T-Coffee.....26
Figura 3.5: Matriz de substituição BLOSUM62.....27
Figura 3.6: Operador de cruzamento produzindo novo alinhamento.....28
Figura 4.1: Esquema geral de um algoritmo evolucionário.....30
Figura 4.2: Alteração de comportamento no cálculo de probabilidade de seleção.....37
Figura 4.3: Pseudocódigo para algoritmo de seleção por roleta fonte: adaptado de (EIBEN; SMITH, 2007).....38
Figura 4.4: Pseudocódigo para algoritmo de seleção por torneio de μ pais fonte: adaptado de (EIBEN; SMITH, 2007).....38
Figura 4.5: Operador one-point crossover.....40
Figura 4.6: Operador n-point crossover.....40
Figura 4.7: Operador uniform crossover.....40
Figura 4.8: Exemplo de operador aritmético de cruzamento.....41
Figura 4.9: Fluxograma geral do EP.....43
Figura 4.10: Comparativo entre as dispersões das curvas de distribuição: Curva de Gauss, em azul, e curva de Cauchy, em vermelho.....45
Figura 4.11: Fluxograma geral do PSO.....49
Figura 5.1: Síntese aplicada aos polos wp e wq produzindo dois polos sínteses, wu e wv. Ambas sínteses wu e wv possuem características de ambos polos, wp e wq.....53
Figura 5.2: Fluxograma básico do ODM.....57
Figura 6.1: Scores obtidos pelo ODM em todas as abordagens e utilizando duas gerações para cada um dos 750 períodos históricos.....73
Figura 6.2: Scores obtidos pelo GA em todas as abordagens.....74
Figura 6.3: Comparativo entre abordagens com diferentes números de gerações por cada período histórico.....78

SUMÁRIO

1 INTRODUÇÃO....1

1.1 Objetivos....2
1.2 Organização do trabalho....3

2 CONCEITOS BÁSICOS EM BIOINFORMÁTICA....4

2.1 Genética e Genômica....4
2.2 DNA, RNA e Expressão Gênica....5
2.3 Proteínas....11
2.4 Sequenciamento....14
2.5 Proteômica....14
2.6 Bioinformática....15

3 ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS....16

3.1 Introdução....16
3.2 Classificação dos métodos de alinhamento múltiplo....19
3.2.1 Algoritmos exatos....19
3.2.2 Algoritmos Progressivos....20
3.2.3 Algoritmos iterativos....22
3.3 Principais algoritmos de alinhamento múltiplo....22
3.3.1 MSA....23
3.3.2 DCA e OMA....23
3.3.3 ClustalW....24
3.3.4 T-Coffee....25
3.3.5 SAGA....27

4 PRINCÍPIOS DE COMPUTAÇÃO EVOLUCIONÁRIA....29

4.1 Visão geral de um algoritmo evolucionário....29
4.1.1 Representação....31
4.1.2 Função de Objetivo....31
4.1.3 População....31
4.1.4 Mecanismo de Seleção de Pais....32
4.1.5 Operadores de Variação....33
4.1.6 Mecanismo de Seleção por Sobrevivência (Substituição)....33
4.1.7 Inicialização....34
4.1.8 Condição de Término....34
4.2 Algoritmos Genéticos....35
4.2.2 Mecanismos para Seleção de pais....36
4.2.3 Operadores de Variação....38
4.2.4 Seleção por sobrevivência....41
4.2.5 Comentários....41
4.3 Programação Evolucionária....42
4.3.1 Representação....42
4.3.2 Seleção de pais e Seleção de Sobrevivência....42
4.3.3 Mutação....43
4.4 Otimização por Enxame de Partículas....46

5 OTIMIZAÇÃO DIALÉTICA....50

5.1 Introdução....50
5.2 Definição Geral....51
5.3 Algoritmo de busca e otimização....54

6 ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS USANDO OTIMIZAÇÃO DIALÉTICA E ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS USANDO ALGORITMOS GENÉTICOS....58

6.1 Base de dados utilizada....60
6.2 Modelagem....61
6.2.1 Candidatos à solução....62
6.2.2 Definição da quantidade de lacunas utilizada....63
6.2.3 Função objetivo....64
6.3 Experimentos....69
6.3.1 Parametrização do ODM....70
6.3.2 Parametrização do GA....70
6.3.3 Abordagem Canônica....71
6.3.4 Abordagem NLC: Número de Lacunas do Clustal....71
6.3.5 Abordagem NFL: Número de Lacunas Flutuante (Variável)....72
6.4 Resultados....73
6.4.1 Análise estatística....74
6.4.2 Resultados com algoritmo dialético discreto com três gerações por período histórico....78
6.5 Conclusão e Comentários....78

7 CONCLUSÕES E PERSPECTIVAS....79

7.1 Contribuições deste trabalho....80
7.2 Dificuldades encontradas....81
7.3 Trabalhos Futuros....81

1 INTRODUÇÃO

Durante o último século, profundos avanços ocorreram na área genética. Os conceitos abordados pelo princípio da herança genética, descritos por Mendel, e as descobertas dos cromossomos e genes como unidades fundamentais para a hereditariedade, por Thomas Morgan, foram extensamente ampliados ao longo deste período sobretudo pela maior oferta de informações biomoleculares. Grande parte desta extensão está relacionada com o desenvolvimento dos primeiros métodos de sequenciamento de genes e proteínas (SANGER; COULSON, 1975), responsáveis por descrever a estrutura de biomoléculas por meio de sequências, e das primeiras técnicas de alinhamento, responsáveis pela extração de informações a respeito das relações entre tais sequências.

Criado por Needleman e Wunsch (NEEDLEMAN; WUNSCH, 1970), o primeiro método de alinhamento desenvolvido faz uso de um algoritmo chamado programação dinâmica para encontrar de forma direta um alinhamento ótimo entre duas sequências biomoleculares. Porém, um alinhamento múltiplo de sequências, ou seja, a generalização desse método para um número qualquer de sequências, é inviável devido ao alto custo computacional associado (NEEDLEMAN; WUNSCH, 1970). Por tal motivo, algoritmos de alinhamento múltiplo adotam heurísticas para a busca por soluções aproximadas.

Na literatura são apresentadas diversas heurísticas que definem diferentes estratégias usadas na busca pelo alinhamento ótimo ou sub-ótimo. A classificação dos algoritmos de alinhamento múltiplo de sequências, ou algoritmos de MSA (*Multiple Sequence Aligment*), segue a estratégia utilizada e suas principais classes são as dos algoritmos exatos, dos progressivos, dos iterativos. A existência de tantos métodos é justificada pelos diferentes resultados obtidos por cada um devido às suas forças e fraquezas individuais.

A combinação de um algoritmo de MSA que produza boas respostas rapidamente, como algum dos algoritmos da família Clustal (LARKIN *et al.*, 2007; SIEVERS *et al.*, 2011), com

métodos de otimização baseados em computação evolucionária consolidados na literatura como algoritmos genéticos ou através do método dialético de otimização, que são capazes de refinar boas soluções já conhecidas, é investigada neste trabalho. A possibilidade de obtenção de diversas soluções ótimas ou sub-ótimas, caso existam, é outro desejado aspecto que é provido por esse tipo de algoritmo iterativo devido à sua natureza estocástica. A utilização de tais métodos iterativos de otimização requer a definição de uma função objetivo capaz de avaliar numericamente o quão adequado à solução do problema é cada alinhamento encontrado ao longo do processo iterativo. Por tal razão, este trabalho sugere uma função objetivo baseado-se em um modelo simples de pontuação de correspondências identificadas em cada coluna, porém, levando-se em conta aspectos biológicos mais relevantes como a distribuição das colunas com alto número de correspondências.

A seguir, na seção 1.1 são descritos o objetivo principal e os objetivos secundários enquanto que a organização deste trabalho é apresentada na seção 1.2.

1.1 OBJETIVOS

Este trabalho tem por objetivo principal propor um método para alinhamento múltiplo de sequências o qual é feito através da aplicação do método dialético de otimização (DOS SANTOS; DE ASSIS, 2009) a soluções encontradas pelo algoritmo para alinhamento múltiplo de sequências ClustalW (LARKIN *et al.*, 2007). Os objetivos específicos deste trabalho são:

1. Desenvolver uma ferramenta de MSA baseada em algoritmos genéticos para ajudar na modelagem da função objetivo sugerida e para produzir dados para realizar comparações com o método dialético de otimização;
2. Propor uma abordagem discreta para o método dialético de otimização (ODM), pois em sua definição são utilizadas componentes de espaço contínuo;
3. Sugerir uma nova representação para os candidatos à solução baseada nas posições das lacunas
4. Sugerir também uma nova função objetivo que seja capaz de identificar aspectos biológicos

desejáveis aos alinhamentos;

1.2 ORGANIZAÇÃO DO TRABALHO

A estrutura desse trabalho, além da primeira parte introdutória, contém outros cinco capítulos. No capítulo 2 são apresentados conceitos biológicos básicos necessários para compreensão deste trabalho. O capítulo 3 apresenta um resumo sobre os principais métodos de alinhamento múltiplo de sequências identificados durante a revisão bibliográfica. No capítulo 4 são descritos conceitos gerais como também específicos da computação evolucionária como uma família de métodos de otimização entre os quais são apresentados os algoritmos genéticos, a programação evolucionária e a otimização por enxame de partículas. O capítulo 5 aborda os conceitos relacionados ao método dialético de otimização enquanto que no capítulo 6 são descritos a metodologia usada e os resultados obtidos. Por fim, no capítulo 7, uma conclusão sobre o trabalho é apresentada seguida de comentários a respeito das dificuldades encontradas e sugestões de trabalhos futuros.

2 CONCEITOS BÁSICOS EM BIOINFORMÁTICA

Neste capítulo, serão apresentados alguns conceitos usados em bioinformática e importantes para o entendimento deste trabalho. Na seção 2.1 são abordados conceitos relacionados a genética e genômica e, na seção 2.2, são discutidos assuntos relativos a DNA e RNA. A seguir, questões relativas a proteínas e proteômica serão abordadas na seção 2.3. A seção 2.4 é dedicada aos conceitos relacionados ao sequenciamento e, por fim, uma explanação geral sobre bioinformática é realizada na seção 2.6.

2.1 GENÉTICA E GENÔMICA

Genética é um termo que se refere ao estudo dos genes e suas funções no processo de herança, ou seja, o meio pelo qual características ou condições são passadas de uma geração à outra. Na genética, os genes são considerados a unidade fundamental da hereditariedade pois transportam as instruções para construção das proteínas que são as responsáveis por definir a atividade das células e todas as funções do organismo. O conjunto completo dessas informações é chamado genótipo do organismo enquanto que fenótipo é o conjunto de características apresentadas pela exposição deste genótipo a um determinado ambiente. Os genes podem assumir várias formas diferentes onde cada um delas é denominada alelo.

Em 1865, Gregor Mendel (ver Figura 2.1) apresentou os resultados do seu estudo sobre características genéticas de plantas de ervilha no qual ele notou que no vegetal estudado haviam dois alelos para cada gene mas apenas um fenótipo era manifestado. Neste trabalho, Mendel descobriu que genes com alelos iguais, chamados de homozigotos, sempre apresentavam um mesmo fenótipo enquanto que genes com alelos diferentes, chamados de heterozigotos, poderiam expressar um dos dois fenótipos representados por tais alelos. Além disso, em seus experimentos Mendel percebeu a existência de uma dominância de alguns fenótipos em relação a outros. Assim, foram concebidas as duas Leis da Genética conhecidas pelas Leis de Mendel. A primeira lei afirma

que, para um determinado gene, a probabilidade de um filho receber um alelo dentre os dois alelos de cada um dos dois pais é a mesma. A segunda Lei afirma que a herança de alelos de um gene em particular ocorre de forma independe aos demais genes. Os genes são distribuídos ao longo do DNA em estruturas conhecidas como cromossomos.

Figura 2.1: Gregor Johann Mendel
(1822– 1884)
Fonte: (PELLINI, [S.d.])

Genômica é um termo recente que descreve o estudo de cada gene contido no [1]genoma de um organismo e suas iterações com outros genes e com o ambiente. Para isso, a genômica faz uso de ferramentas como métodos de sequenciamento de DNA e técnicas da bioinformática para sequenciar, montar e analisar a estrutura dos genomas.

2.2 DNA, RNA E EXPRESSÃO GÊNICA

O ácido desoxirribonucleico, ou DNA, é uma molécula que codifica as informações genéticas que coordenam o funcionamento e desenvolvimento de todos os seres vivos e de alguns vírus.

[1] Genoma de um organismo é o conjunto formado por todos os genes deste organismo.

Do ponto de vista químico, uma molécula de DNA é uma dupla de fitas, dispostas em hélice, composta de dois longos polímeros formados por unidades simples, monômeros, conhecidas por nucleotídeos. Os nucleotídeos são compostos orgânicos formados por uma base nitrogenada (guanina, adenina, timina ou citocina)[2], um açúcar de cinco carbonos e ao menos um grupo fosfato. Em cada um destes longos polímeros, dois nucleotídeos são ligados entre si através do grupo fosfato de um nucleotídeo com o açúcar do outro. Como efeito, a estrutura de cada cadeia é definida por esqueleto composto de moléculas alternadas de grupos de açúcar e fosfato, e a tal esqueleto são fixadas nos açúcares as bases nitrogenadas de forma que, em cada posição, uma purina (guanina ou adenina) de uma das cadeias sempre é acompanhada de uma pirimidina (timina ou citocina) na cadeia oposta. Além disso, sentidos da construção das duas cadeias de nucleotídeos são opostos entre si. Assim, uma molécula (ou segmento) de molécula de DNA é representada por uma sequência formada por letras de um alfabeto composto pelas iniciais de cada uma das quatro bases nitrogenadas (A,G,T e C). A estrutura química da molécula de DNA com as bases complementares ligadas por pontes de hidrogênio pode ser vista na Figura 2.2.

As informações codificadas no DNA são utilizadas principalmente na construção de proteínas e de RNA os quais, juntos com o DNA, são as três principais macromoléculas essenciais a todas as formas de vida conhecidas. As proteínas são construídas a partir de uma cadeia de aminoácidos que é montada através de um mecanismo conhecido por síntese proteica. A estrutura química do RNA é bastante similar à do DNA, mas se difere em três pontos: (i) como o DNA, a molécula do RNA é composta de um polímero de nucleotídeos, porém disposto em uma única cadeia; (ii) o açúcar do RNA contém uma ribose em vez de uma desoxirribose; (iii) a base complementar à adenina não é a timina e sim a uracila. A estrutura química da molécula de RNA também é apresentada na Figura 2.2.

[2] Na representação de uma base nitrogenada é usada a primeira letra do nome da base.

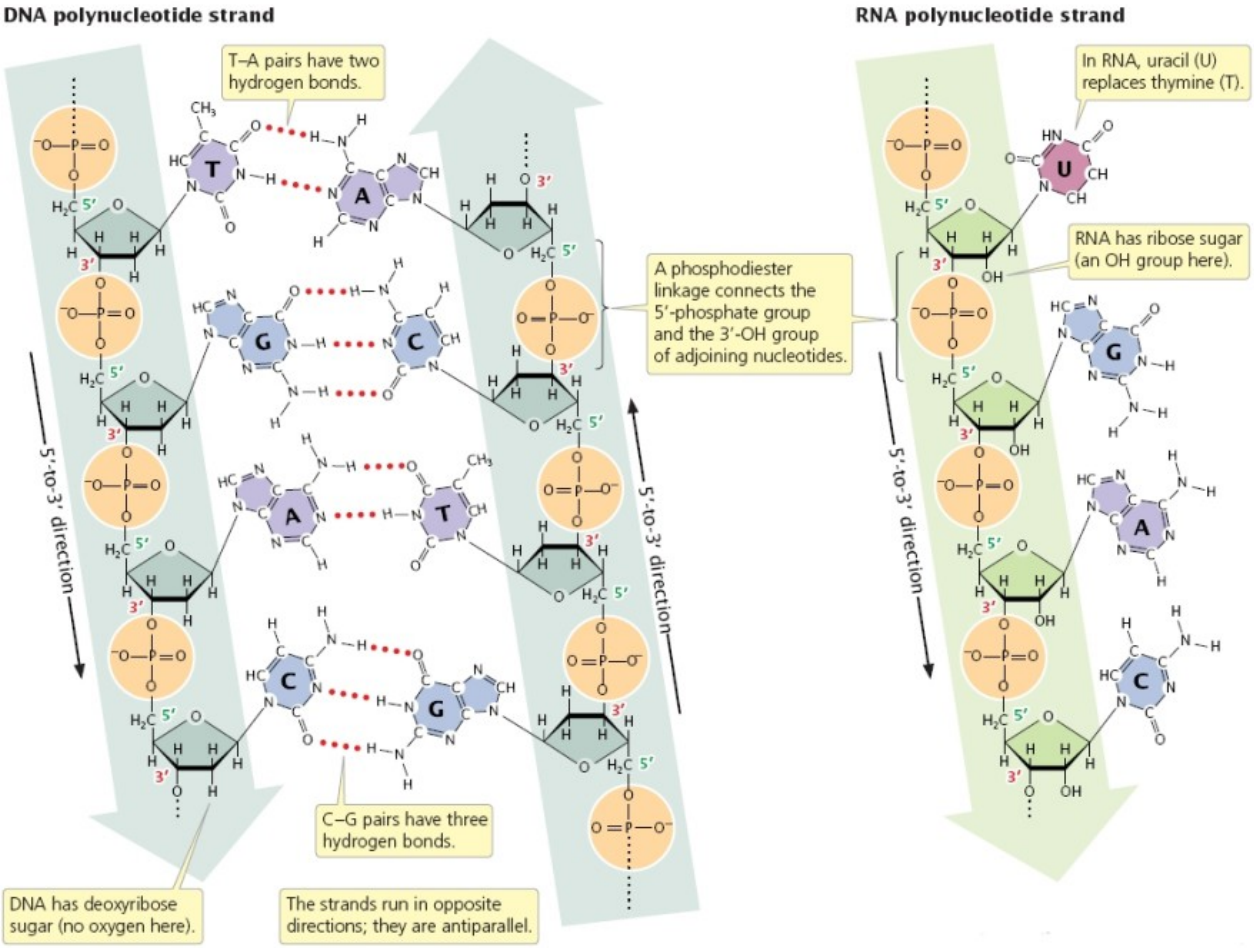

Figura 2.2: Estruturas das moléculas de DNA (à esq.) e RNA (à dir.)
fonte: (CLANCY, 2014)

Ao longo da enorme cadeia de DNA existem regiões que não codificam sequências de proteínas. Em genética, tais regiões são chamadas de DNA não codificante. Algumas porções de DNA não codificante são transcritas em moléculas funcionais de RNA não codificante, como RNA transportador e RNA ribossomal[3], enquanto outras não são transcritas ou transcrevem RNA de função desconhecida. A quantidade de DNA genômico e de DNA não codificante e a proporção entre estes varia bastante entre os organismos. Por exemplo, em casos extremos tem-se o percentual de DNA não codificante de 98,5% para o *Homo Sapiens* (ELGAR; VAVOURI, 2008) e de apenas 3% para a *Utricularia gibba* (IBARRA-LACLETTE *et al.*, 2013), uma pequena planta carnívora aquática que se alimenta de pequenos invertebrados.

[3] Agem durante síntese proteica.

Os genes são as regiões codificantes do DNA que são processadas em um produto gênico funcional, como proteínas ou RNA, através de um processo chamado expressão gênica. Neste processo, a informação contida no gene, disposta em uma sequência de bases, é lida em agrupamentos de três bases cada chamados códons. Cada códon corresponde a um dos 20 aminoácidos existentes seguindo o código genético apresentado na Tabela 2.1. Por fim, o aminoácido correspondente ao códon lido é inserido na sequência. O processo se repete até que a sequência que dará origem à proteína ou ao RNA em questão seja construída.

Tabela 2.1: Código Genético

1ª. Base do Códon		2ª. Base do Códon								3a'Base do Códon
		U		C		A		G		
	U	**UUU**	Fenilamina (Fen)	**UCU**	Serina (Ser)	**UAU**	Tirosina (Tir)	**UGU**	Cisteína (Cis)	U
		UUC		**UCC**		**UAC**		**UGC**		C
		UUA	Leucina (Leu)	**UCA**		***UAA***	Códões de iniciação	***UGA***	Códão de finalização	A
		UUG		**UCG**		***UAG***		**UGG**	Triptofano (Trp)	G
	C	**CUU**	Leucina (Leu)	**CCU**	Prolina (Pro)	**CAU**	Histidina (His)	**CGU**	Arginina (Arg)	U
		CUC		**CCC**		**CAC**		**CGC**		C
		CUA		**CCA**		**CAA**	Glutamina (Glu)	**CGA**		A
		CUG		**CCG**		**CAG**		**CGG**		G
	A	**AUU**	Isoleucina (Ile)	**ACU**	Treonina (Tre)	**AAU**	Asparagina (Asn)	**AGU**	Serina (Ser)	U
		AUC		**ACC**		**AAC**		**AGC**		C
		AUA		**ACA**		**AAA**	Lisina (Lis)	**AGA**	Arginina (Arg)	A
		AUG	Metionina (Met) Códão de iniciazação	**ACG**		**AAG**		**AGG**		G
	G	**GUU**	Valina (Val)	**GCU**	Alanina (Ala)	**GAU**	Ácido aspártico (Asp)	**GGU**	Glicina (Gli)	U
		GUC		**GCC**		**GAC**		**GGC**		C
		GUA		**GCA**		**GAA**	Ácido glutamínico (Glu)	**GGA**		A
		GUG		**GCG**		**GAG**		**GGG**		G

A síntese de proteica é um exemplo de expressão gênica cujos produtos gênicos construídos são proteínas sendo descrita em duas etapas: a transcrição, que tem como seu papel fundamental a

produção de cópias na forma de RNA mensageiro, ou mRNA, do segmento de DNA lido, e a tradução, onde a informação transcrita do DNA, sob a forma de mRNA, é traduzida para indicar a ordem correta dos aminoácidos que irão compor a cadeia polipeptídica da proteína. Um esquema geral da síntese proteica é exibido na Figura 2.3.

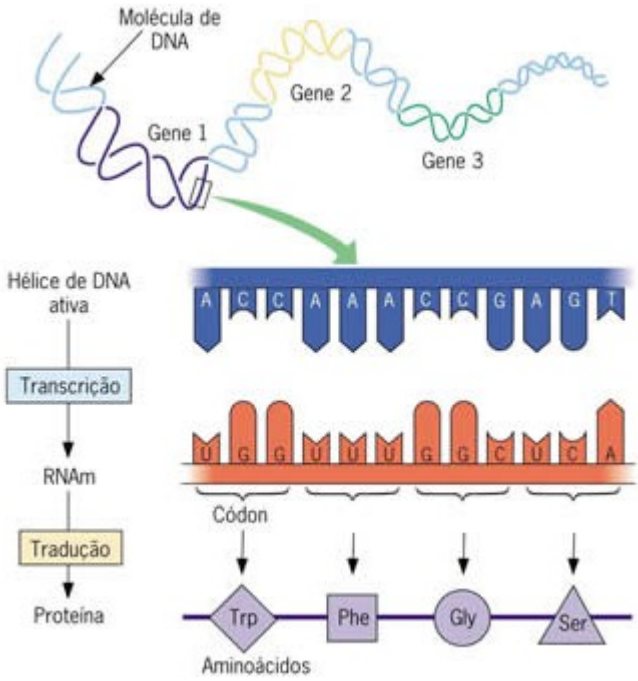

Figura 2.3: Esquema geral da expressão gênica na produção de proteínas
fonte:(“SÓ BIOLOGIA,” [S.d.])

Na transcrição ocorre a síntese do RNA mensageiro, ou mRNA, a partir de uma cadeia de DNA. Este processo é iniciado quando a enzima RNA polimerase se conecta à molécula de DNA fazendo que esta se abra longo de um pequeno segmento separando as cadeias. Esta posição onde ocorre tal ligação é determinada por uma sequência específica a qual é chamada de promotor. Neste ponto, os nucleotídeos livres na célula começam a se parear com o segmento aberto da fita de DNA. Tal processo continua até que seja identificada na fita de DNA uma sequência de marcação de término a qual provoca o desligamento entre o mRNA sintetizado e a enzima RNA polimerase. A etapa de transcrição pode ser visualizada na Figura 2.4.

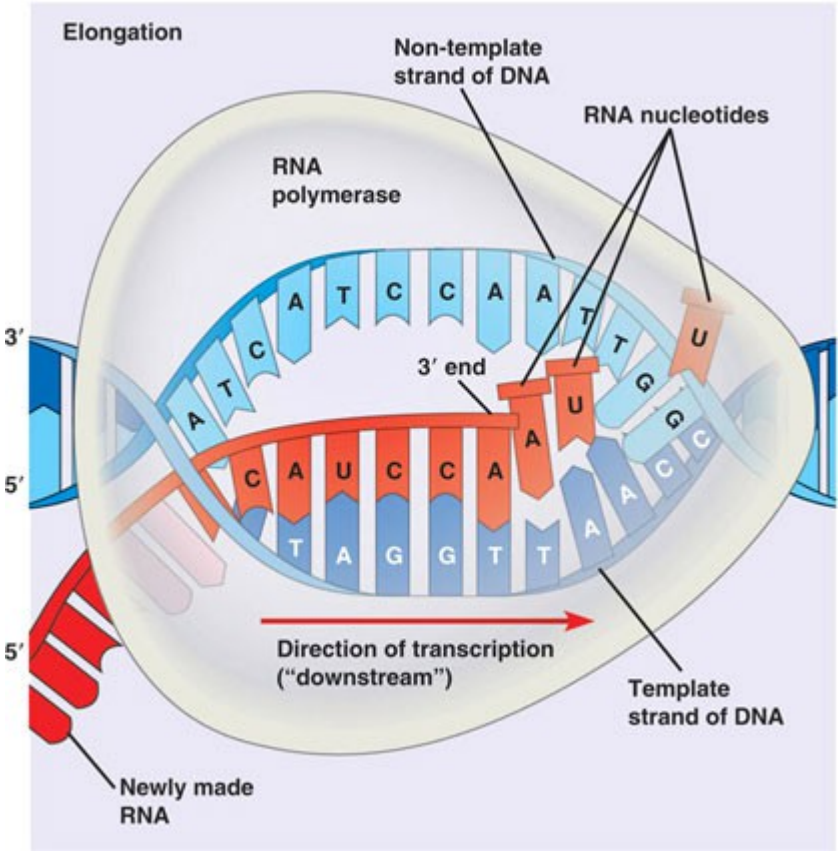

Figura 2.4: Processo de Transcrição de DNA em mRNA
fonte:(KEMCCALLUM, 2011)

As moléculas de mRNA sintetizado possuem as informações para síntese da proteína. Tal informação é codificada em agrupamentos de três bases consecutivas, chamados códons. A cada códon é associado um dentre os 20 aminoácidos existentes. No início da tradução, o RNA ribossomal ou ribossomo ou rRNA, é ligado ao mRNA e desliza ao longo da molécula até encontrar uma sequência que corresponda ao códon de marcação de início (AUG). Após isso, um tRNA recolhe aminoácidos correspondentes ao próximo códon, levando-os até o ribossomo e este anexa o códon fornecido pelo tRNA à cadeia de aminoácidos em construção. Em seguida o ribossomo libera o tRNA, desloca-se sobre o mRNA e aguarda a chegada de outro tRNA que traga o códon associado à nova posição atual. Este processo se repete até que o ribossomo identifica um códon de terminação e desliga-se da molécula de mRNA. Após isso, a cadeia polipeptídica sintetizada desliga-se do ribossomo e é dobrada através das interações dos aminoácidos presentes na cadeia

ganhando um formato tridimensional característico o qual participará da definição de sua função no organismo.

2.3 PROTEÍNAS

Proteínas são macromoléculas que consistem de um ou mais cadeias de resíduos de aminoácidos e executam uma grande lista de funções dentro dos organismos vivos (KARP, 2008) sendo amplamente usadas no estudo da história evolucionária dos organismos.

As proteínas possuem quatro tipos de estruturas: primária, secundária, terciária ou quaternária. A estrutura primária é definida pela sequência de aminoácidos que formam proteína e pelas pontes de dissulfureto. Esta sequência determina a estrutura dos outros níveis mais altos da molécula.

A estrutura secundária é definida pelo arranjo espacial formado pelas dobras na estrutura primária provocadas pelas interações moleculares entre os aminoácidos próximos entre si na sequência primária da proteína. Um arranjo possível é o alfa-hélice, onde a cadeia polipeptídica é enrolada sobre um eixo imaginário e é estabilizada por pontes de hidrogênio entre os componentes da ligação peptídea. Outro padrão de forma identificada é o beta folha, no qual os aminoácidos adotam o formato de uma folha de papel por meio de consecutivos filamentos lineares que são alinhados lado a lado. Tal estrutura é estabilizada através de pontes de hidrogênio entre aminoácidos dos filamentos vizinhos. Por fim, outras partes que não são altamente estáveis adotam uma forma de bobina aleatória ou *coil*. A Figura 2.5 mostra formas possíveis de estruturas secundárias de uma proteína: a espiral é uma alfa-hélice, os filamentos lado a lado formam uma beta-folha e os demais trechos cujas formas não seguem um padrão uniforme são estruturas do tipo *coil* ou *random coil*.

A estrutura terciária é definida pela forma tridimensional completa da molécula. Neste nível de estrutura, aminoácidos distantes em relação à estrutura primária podem estar próximos no nível de estrutura terciária devido à forma como a cadeia se dobra.

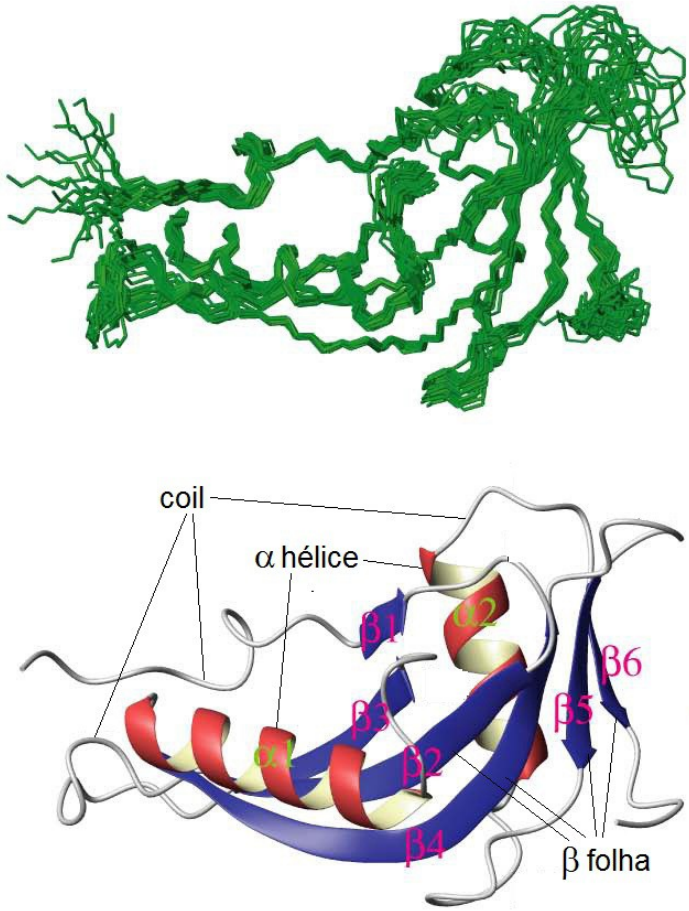

Figura 2.5: Formas de estrutura secundária: alfa hélice, beta folha e bobina aleatória
fonte:(ROTHAMSTED, [S.d.])

Por fim, a estrutura quaternária é o arranjo entre duas ou mais subunidades polipeptídicas da proteína. Na Figura 2.6 são apresentados os quatro níveis de estrutura de uma proteína e as pontes de hidrogênio que estabilizam as forma da alfa-hélice e da beta-folha.

É esperado que proteínas de espécies diferentes, porém originadas de um mesmo ancestral, possuam diferenças entre suas estruturas primárias. Apesar disso, existem regiões da cadeia que executam forte influência na definição da estrutura secundária de um trecho específico da proteína. Estes segmentos, conhecidos por *motifs* ou *domains*, possuem baixa probabilidade de sofrer mutação, ou seja, correspondem a regiões de alta conservação das bases pois alterações mínimas em sua estrutura normalmente causam impactos significativos na capacidade da proteína em realizar sua função biológica. Essa é uma forte motivação para que proteínas sejam usadas em estudos de

história evolucionária de organismos. Além da participar de funções enzimáticas quando se apresentam sob a forma de enzimas, as proteínas ainda podem realizar funções de regulação (quando se apresentam como vitaminas), de reserva (quando se apresentam na forma de sementes), de transporte (caso transportem nutrientes e metabólicos entre tecidos e fluidos), estrutural (quando compõe matéria-prima para construção de estruturas celulares), de defesa (quando agem contra antígenos), hormonal e outras (SCITABLE BY NATURE EDUCATION, [S.d.]).

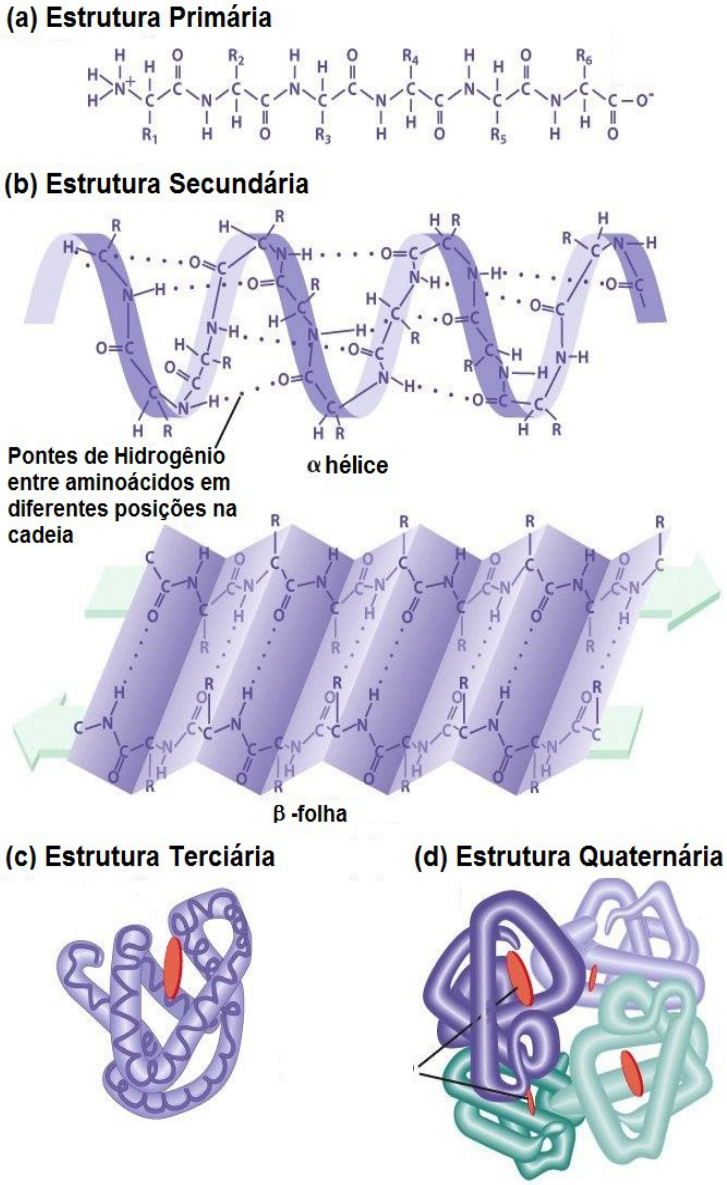

Figura 2.6: Níveis de Estruturas de Proteínas
fonte:(LAMORTE, 2013)

2.4 SEQUENCIAMENTO

O sequenciamento de um gene é processo pelo qual é determinada a cadeia de nucleotídeos que o compõe. Um genoma é muito extenso para ser sequenciado inteiramente pois pode possuir bilhões de bases. Assim, primeiramente o genoma é dividido em pequenos segmentos os quais são sequenciados individualmente e, após isso, ordenados de forma que seja construída uma única sequência que corresponderá ao sequenciamento completo do genoma inicial. Esta fragmentação do genoma é normalmente feita através da estratégia *shotgun* (VENTER, 1998) na qual o DNA é submetido a altas taxas de vibração que promovem a quebra da cadeia em vários fragmentos que são geralmente únicos. Em seguida, é iniciado o sequenciamento das bases de cada um dos fragmentos através de métodos como o método de Sanger (SANGER; COULSON, 1975), conhecido como terminação de cadeia, e o método chamado pirosequenciamento (NYRÉN, 2007; RONAGHI, 1998; RONAGHI *et al.*, 1996).

As sequências resultantes podem possuir regiões de baixa qualidade, baixa complexidade, ou que não fazem parte do organismo de origem do genoma em questão. Estas regiões, chamadas de artefatos, possuem mais chances de conter erros de sequenciamento. Normalmente tais problemas são detectados e eliminados em uma etapa de pós-processamento ao término de um sequenciamento.

2.5 PROTEÔMICA

Uma vez que o sequenciamento seja concluído, é possível descobrir informações a respeito de quais regiões do genoma sequenciado codificam proteínas e quais são as proteínas codificadas. O conjunto das proteínas expressas por um genoma em determinadas condições de tempo, espaço, estado patológico e estímulos externos é chamado de proteoma.

O proteoma, diferentemente do genoma, é variável e se altera de acordo com as condições as quais o organismo está exposto. A análise do proteoma de um organismo visando identificar,

quantificar e estudar tais modificações ocorridas com as proteínas após o período de tradução é chamada proteômica. Devido principalmente ao avanço tecnológico no estudo de proteínas e a crescente e contínua geração de dados genômicos, a proteômica tornou-se uma ferramenta bastante útil para entender a genética e a fisiologia de diversos organismos (CÁNOVAS *et al.*, 2004; RAMPITSCH; BYKOVA, 2009). A combinação de técnicas como a eletroforese bidimensional, que é capaz de separar diferentes proteínas sintetizadas por uma célula, com as ferramentas provenientes da bioinformática permitiu um maior entendimento a respeito de uma grande quantidade de proteínas.

2.6 BIOINFORMÁTICA

A bioinformática é definida como uma área científica interdisciplinar que desenvolve métodos para armazenamento, recuperação, organização e análise de dados biológicos. Seu maior papel é desenvolver ferramentas em *software* para produzir informações relevantes a partir destes dados.

3 ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS

3.1 INTRODUÇÃO

O alinhamento múltiplo de sequências (*Multiple Sequence Alignment*, MSA) é um conjunto de técnicas utilizadas para inferir informações biológicas de um conjunto de sequências (SIMOSSIS; KLEINJUNG; HERINGA, 2003). É considerada a tarefa mais comum e mais importante da bioinformática (NOTREDAME, 2002) pois pode fornecer informações ricas sobre estrutura e função de genes e proteínas.

Um MSA é produzido pelo alinhamento de mais de duas sequências no qual deseja-se identificar as posições (colunas) onde as bases de nucleotídeos são equivalentes. Visualmente, um MSA é normalmente apresentado como uma matriz de duas dimensões na qual as sequências são as linhas e as colunas são formadas de bases e lacunas.

Em um MSA, normalmente as colunas são constituídas de bases equivalentes. Tal equivalência é produzida pelo deslocamento horizontal de bases através da inserção de espaços em branco (lacunas) em posições apropriadas de forma que a relação biológica das sequências seja melhor representada (SIMOSSIS; KLEINJUNG; HERINGA, 2003). A presença de lacunas está relacionada com a ocorrência de mutações como exclusão ou inserção de bases.

Um exemplo de alinhamento múltiplo de sequências de DNA pode ser visto na Figura 3.1 (GALDINO *et al.*, 2010) onde a exclusão de bases, associada à ocorrência de poucas lacunas na mesma coluna, é destacada por uma seta enquanto que a inserção de bases, associada à ocorrência de diversas lacunas em uma mesma coluna, é destacada com fundo em cinza.

O MSA possui grande relevância na bioinformática pois, através dele, é possível obter informações de aspecto evolucionário a respeito de relações estruturais e funcionais existentes entre as sequências envolvidas, sejam elas de proteínas, DNA ou RNA (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003). Tais informações são fundamentais na busca por

respostas relacionadas a análises de homologia entre sequências como a filogenética, a identificação de padrões conservados ([4]*mofits* e domains) e a predição de estruturas secundária e terciária de proteínas (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003; THOMPSON, J D; PLEWNIAK; POCH, 1999; THOMPSON, JULIE D *et al.*, 2011).

Figura 3.1: Exemplo de alinhamento múltiplo de sequências de proteínas
fonte:(GALDINO *et al.*, 2010)

A literatura mostra que a construção de MSAs normalmente segue modelos estatísticos (matrizes de similaridades) (NOTREDAME, 2002). Isso permite ao MSA ser visto como modelo para tratar hipóteses evolucionárias onde o objetivo principal é obter a melhor estimativa da história evolucionária das sequências alinhadas.

Apesar de seu contínuo desenvolvimento, as complicadas relações biológicas existentes entre sequências homólogas, ou seja, sequências que provém de organismos de espécies diferentes mas que possuem um ancestral em comum, combinadas com a falta de informação sobre suas verdadeiras histórias evolucionárias fazem da exatidão do MSA algo difícil de ser garantido (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003).

Diante de tais particularidades, o MSA é considerado um dos problemas candentes da Bioinformática, tanto do ponto de vista das complexidades (computacional e biológica) envolvidas

[4] *Motifs* são pequenas regiões contínuas, com comprimento de 3 a 9 bases, frequentemente envolvidos na função ou na integridade estrutural de uma proteína enquanto que *domains* são vistos como unidades de blocos de bases recorrentes durante a evolução molecular. Fonte: http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd_help.shtml.

na construção dos métodos de alinhamento, quanto em relação à importância da pesquisa genética para o contexto atual.

A realização de um alinhamento múltiplo de sequências envolve a solução de três questões técnicas distintas (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003): (i) a escolha correta das sequências, (ii) a escolha de uma função objetivo capaz de qualificar biologicamente um alinhamento e (iii) a computação de um alinhamento ótimo (obtido através da maximização do *score*).

A escolha das sequências é guiada pelo grau de similaridade existente entre elas. Alguns métodos de alinhamento múltiplo exigem a escolha de sequências de alta similaridade. Isso ocorre porque o uso de sequências distantes em similaridade normalmente insere, ruído que acaba desviando o algoritmo de alinhar os resíduos realmente relevantes (NOTREDAME, 2002). Apesar de tal restrição ser um pouco suavizada em outros métodos, independente da escolha das sequências, sempre será produzido um alinhamento e, portanto, cabe aos biólogos confirmarem se tal alinhamento tem ou não significado biológico (NOTREDAME, 2002). A escolha de sequência é portanto, uma tarefa difícil. Por isso, são utilizadas ferramentas de busca, como os programas BLAST, para identificar conjuntos de sequências homólogas em base de dados públicas (ALTSCHUL *et al.*, 1990). Porém, como tais ferramentas utilizam modelos estatísticos para aproximar a realidade biológica, sempre existe a probabilidade da homologia ser mal representada pela similaridade fazendo com que sequências inadequadas sejam incorporadas ao alinhamento (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003).

Para o problema de MSA uma função objetivo busca mensurar a qualidade biológica de um alinhamento e sua definição é uma tarefa não-trivial. Para Notredame (2002), uma função objetivo perfeita para uma ferramenta de MSA deve, teoricamente, conter todas as informações a respeito das sequências, incluindo sua estrutura, função e evolução histórica.

Frequentemente é usada uma função objetivo baseada em um modelo de pontuação

conhecido por *sum of pairs* (NEEDLEMAN; WUNSCH, 1970) a qual leva em conta as correspondências entre bases de uma mesma coluna e a atribuição de penalidades para inserções e exclusões de bases. A computação da pontuação total, chamada de *score*, é obtida pela soma entre a pontuação total, obtida pelas correspondências identificadas em cada coluna, e a penalidades atribuídas às lacunas utilizadas (*gaps*).

Independente do desafio biológico fundamental, o MSA é um problema que demanda grande quantidade de recursos computacionais. Realizar um alinhamento múltiplo direto e de forma exata exige a manipulação de uma matriz multidimensional de busca na qual cada sequência representa uma dimensão extra. Realizado dessa forma, o MSA torna-se um problema de complexidade $O(L^N)$ para N sequências de comprimento L tornando-o proibitivo mesmo para um pequeno número de sequências. Assim, as ferramentas existentes são apenas heurísticas que disponibilizam uma solução aproximada para o problema. Muitas dessas heurísticas são baseadas em diferentes paradigmas, cada qual adequado a um determinado conjunto de situações (NOTREDAME, 2002). Dessa forma, os algoritmos de MSA podem ser classificados segundo a heurística utilizada conforme a seguir.

3.2 CLASSIFICAÇÃO DOS MÉTODOS DE ALINHAMENTO MÚLTIPLO

3.2.1 Algoritmos exatos

São algoritmos que empregam programação dinâmica multidimensional. A heurística existente nessa classe de algoritmos limita-se a reduzir a computação através de uma estimação de qual região da matriz diagonal multidimensional de busca deve ser explorada pelo algoritmo exato de alinhamento de pares de sequência Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970). Embora tal heurística seja capaz de obter alinhamentos mais precisos e próximos do ótimo, ela ainda possui grandes limitações de uso devido a quantidade de memória e processamento utilizados. Devido à sua natureza exata, esta classe de algoritmos é limitada em usar exclusivamente a função objetivo *sum-of-pairs*.

Outros métodos baseados nesta heurística foram desenvolvidos para reduzir os custos computacionais (STOYE; MOULTON; DRESS, 1997)(REINERT; STOYE; WILL, 2000) porém mesmo estes não são capazes de manipular grandes massas de dados.

3.2.2 Algoritmos Progressivos

Algoritmos progressivos de alinhamento de sequências constituem uma das formas mais usadas, simples e eficientes de realizar o alinhamento de múltiplas sequências pois fazem uso de pouca memória e pouco tempo para produzir bons alinhamentos (NOTREDAME; HIGGINS; HERINGA, 2000)(HIGGINS; SHARP, 1988a)(CORPET, 1988). Foi inicialmente descrito por Hogeweg e Hesper (HOGEWEG; HESPER, 1984) e depois aprimorado por Feng e Doolittle (FENG; DOOLITTLE, 1987) e Taylor (TAYLOR, 1988).

Sua heurística define um processo progressivo de montagem do alinhamento múltiplo a partir da realização gradual de alinhamentos de pares de sequências até que todas as sequências estejam alinhadas.(FENG; DOOLITTLE, 1987; HOGEWEG; HESPER, 1984; SIMOSSIS; KLEINJUNG; HERINGA, 2003; TAYLOR, 1988). Essa montagem é realizada em três etapas conforme a seguir:

- Na primeira são realizados alinhamentos em pares entre todas as *N* sequências. Os *scores* obtidos de cada um destes alinhamentos são utilizados por alguma técnica de agrupamento (*clustering*) para construir uma matriz diagonal que contém um valor normalizado de similidade entre cada par das *N* sequências.
- Na segunda etapa, as sequências mais similares (menor distância) são agrupadas para formar um dendrograma[5] o qual é usado para guiar o alinhamento da próxima etapa. Um exemplo de dendrograma pode ser visualizado na Figura 3.2.
- Na terceira etapa, sequências e grupos de sequências (formados na etapa anterior) são alinhados até que não haja mais nada para alinhar. Tais grupos de sequências são convertidos

[5]Tipo específicos de diagrama que ilustra graficamente a ordenação hierárquica ascendente de dados

em *profiles* que são estruturas que representam em um alinhamento a frequência dos caracteres para cada coluna.

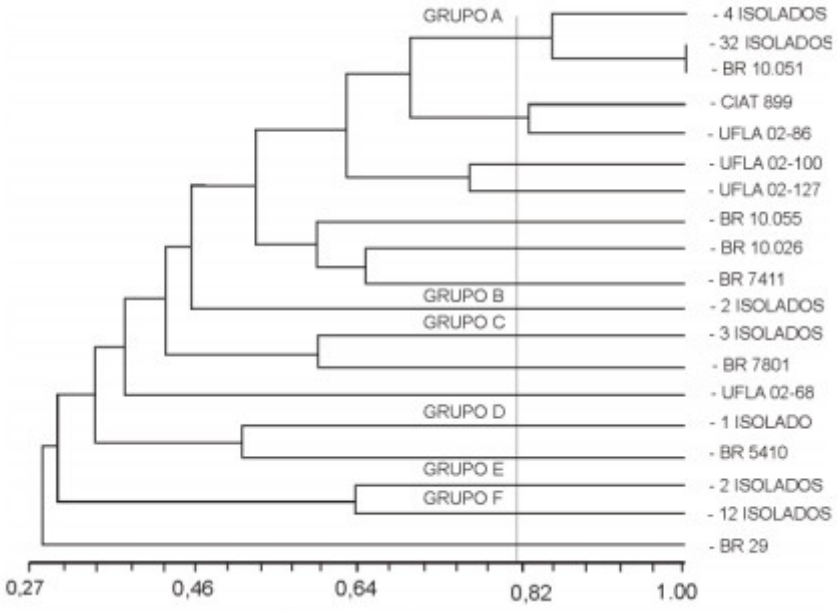

Figura 3.2: Exemplo de dendrograma de similaridade

O programa de MSA de heurística progressiva mais conhecido é o ClustalW (THOMPSON, J D; PLEWNIAK; POCH, 1999) o qual é capaz de alinhar poucas milhares de sequências de comprimento moderado (SIEVERS *et al.*, 2011).

Apesar de possuir uma eficiente e estável estratégia, a heurística progressiva falha por não ser capaz de revisar alinhamentos realizados anteriormente de forma que erros cometidos no início são mantidos até o final do processo de construção e não podem ser reparados (ver Figura 3.3). Para combater esse problema, mais conhecido por "*once a gap always a gap*", em (NOTREDAME; HIGGINS; HERINGA, 2000) foi desenvolvida uma heurística baseada consistência. Tal heurística permitiu o surgimento algoritmos progressivos de precisão superior, na ordem de 5 a 10%, em relação aos antigos algoritmos puramente progressivos (SIEVERS *et al.*, 2011). Porém, tais melhorias estão associadas a um aumento no custo computacional geral limitando tais algoritmos baseados em consistência a operar com uma quantidade de sequências reduzida (CUTELLO *et al.*, 2011; SIEVERS *et al.*, 2011). São considerados algoritmos progressivos baseados em consistência: T-Coffee (NOTREDAME *et al.* 2000) e PRALINE (HERINGA, 1999;

HERINGA, 2002).

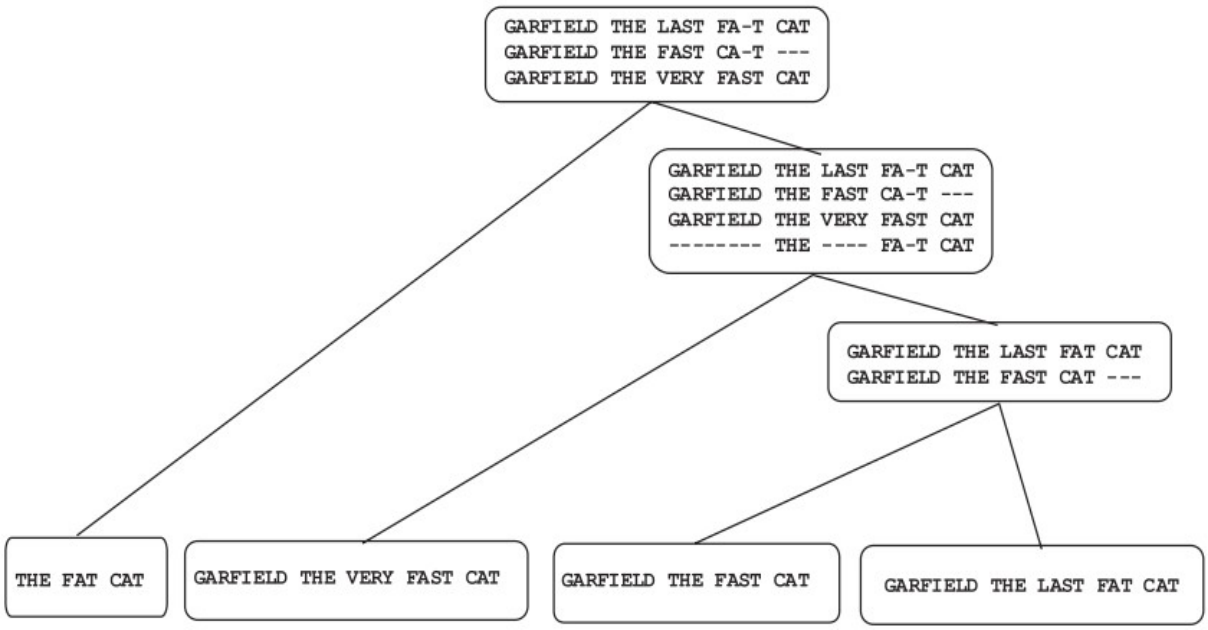

Figura 3.3: Deficiência do método progressivo em não reavaliar lacunas inseridas inicialmente fonte:(NOTREDAME, 2002)

3.2.3 Algoritmos iterativos

Algoritmos interativos buscam otimizar um alinhamento não ótimo previamente conhecido através de algum processo iterativo que produza melhorias até que algum limite pré-definido seja alcançado. As soluções não-ótimas utilizadas neste processo são obtidas por outro algoritmo de alinhamento ou através de uma heurística própria. Algoritmos iterativos podem ser classificados em estocásticos e não-estocásticos. No primeiro caso, algum grau de aleatoriedade é incorporado na solução; algoritmos de alinhamentos que fazem uso de algoritmos genéticos são um bom exemplo de algoritmos iterativos estocásticos. Algoritmos iterativos não-estocásticos normalmente são algoritmos baseados na heurística progressiva contendo, porém, algum processo interno iterativo no qual cada sequência é realinhada ao alinhamento múltiplo até que este processo falhe em promover melhorias ao alinhamento.

3.3 PRINCIPAIS ALGORITMOS DE ALINHAMENTO MÚLTIPLO

O problema do MSA, ou seja, a busca pelo alinhamento ótimo para um determinado

conjunto de sequências, é estudado durante mais de 30 anos e a quantidade de abordagem desenvolvidas que tentam solucioná-lo, cada uma com suas qualidades e fraquezas, ainda continua a crescer. Infelizmente tal diversidade de métodos acaba tornando seu uso difícil para não especialistas identificarem qual método é mais adequada a cada situação. A seguir, alguns dos principais métodos são descritos de forma resumida

3.3.1 MSA

É um algoritmo global exato que portanto emprega programação dinâmica. Também faz uso de alinhamentos locais, e por tal razão é rotulado em (SIMOSSIS; KLEINJUNG; HERINGA, 2003) como um algoritmo simultâneo. Para reduzir computação usada, o algoritmo MSA utiliza uma abordagem desenvolvida por Carrillo e Lipman (CARRILLO; LIPMAN, 1988) que estima, através de alinhamentos de pares, quanto da matriz multidimensional de busca precisa ser percorrida. Apesar de tal aproximação ser capaz de produzir alinhamentos com *score* otimizado e, a princípio, mais precisos e livres de erros que alinhamentos produzidos por métodos progressivos, tal abordagem tem grandes limitações sobre a quantidade de sequências que podem ser manipuladas devido ao seu grande consumo de recursos computacionais (processamento e memória). Segundo (SIEVERS *et al.*, 2011) até 10 sequências de comprimento entre 200 e 300 podem ser alinhadas com o método MSA em tempo aceitável.

3.3.2 DCA e OMA

O DCA é um método de alinhamento exato baseado no método exato MSA, mas que utiliza uma estratégia “dividir para conquistar” na qual as sequências são dividas em subsequências que são pequenas o suficiente para alimentar o algoritmo MSA sem sobrecarregá-lo. A ideia é cortar as sequências nos pontos corretos, de acordo com um limiar pré-definido. Quanto menor o limiar, mais rápido e menos preciso se tornará o alinhamento. Os pequenos conjuntos de subsequências resultantes são alinhados através do MSA e no final são concatenados para produzir o alinhamento

completo final (SIMOSSIS; KLEINJUNG; HERINGA, 2003). Porém, apesar de apresentar melhorias no consumo de recursos computacionais, a quantidade de sequências que o DCA pode alinhar em tempo razoável também é pequena como no MSA.

Existe um algoritmo chamado OMA que é uma variação do DCA implementada com um esquema iterativo. A heurística do OMA é inicializar o DCA com o limiar de comprimento das subsequências com um valor muito pequeno. Assim, na primeira iteração temos um alinhamento pouco preciso porém muito rápido de ser construído. O limiar é então aumentado gradualmente ao longo das iterações produzindo dessa forma alinhamento mais precisos porém consumindo mais recursos computacionais a cada iteração. Além destas, outras técnicas de otimização são utilizadas, inclusive técnicas de paralelismo. Porém, apesar das melhorias, o OMA também é um método que não é capaz de manipular grandes bases de dados.

3.3.3 ClustalW

O ClustalW e sua versão com interface gráfica, ClustalX, são versões atualizadas do algoritmo global progressivo para alinhamento múltiplo Clustal (HIGGINS; SHARP, 1988b) e por muito tempo foram considerados o método padrão ouro para MSA.

O algoritmo progressivo do ClustalW é composto de três etapas distintas. Na primeira etapa do alinhamento múltiplo, é construída uma matriz diagonal $\boldsymbol{M}$ de dimensões N x N, com N igual ao número de sequências, de forma que cada elemento $a_{i,j}$ de $\boldsymbol{M}$ corresponde ao resultado (*score*) do alinhamento em pares formado entre i-ésima e a j-ésima sequência, com $i, j \leq N$. Na segunda etapa, é construído um dendrograma (árvore guia) a partir da matriz $\boldsymbol{M}$ a qual é capaz por identificar quais sequências são mais similares entre e si. Na terceira etapa, seguindo a ordem do dendrograma as sequências são selecionadas para a realização de alinhamentos em pares e, gradativamente o alinhamento múltiplo é construído.

O ClustalW tem melhor precisão quando entre as sequências a serem alinhadas não há sequências discrepantes e possuem alta similaridade.

3.3.4 T-Coffee

O T-Coffee (NOTREDAME; HIGGINS; HERINGA, 2000) é um algoritmo progressivo baseado em consistência para alinhamento múltiplo de sequências. A estratégia deste algoritmo é fazer uso de informações de alinhamentos em pares, globais e locais, para aumentar a precisão do alinhamento múltiplo final e, além disso, evitar os problemas de precisão comuns a métodos progressivos puros.

Inicialmente são construídas duas bibliotecas de alinhamentos, uma para alinhamentos locais e outra para alinhamentos globais. Para cada par de sequência são calculados: um alinhamento global a partir do ClustalW e dez alinhamentos locais, que correspondem aos alinhamentos de maior *score* e sem interseção encontrados pelo Lalign (HUANG; MILLER, 1991). Uma biblioteca primária é construída agrupando-se os dados de ambos alinhamentos. Este agrupamento é realizado junto a um esquema de atribuição de pesos para cada par de resíduo. A biblioteca primária é então expandida para também conter a informação sobre como pares de resíduos alinham com outros resíduos da biblioteca. Isso é feito atribuindo um peso a cada par de resíduos alinhados (ex.: resíduo *x* da sequência *A* alinhado com resíduo *y* da sequência *B*). Este peso reflete o grau de consistência do alinhamento entre este par já alinhado e os resíduos de todas as outras sequências. Assim, esse processo produz trios de pesos que são capazes para aferir o quão bem sequências alinhadas são similares a outras sequências do conjunto de dados, no lugar de apenas investigar de forma isolada a similaridade em pares de sequências. O alinhamento final é então produzido utilizando-se a biblioteca expandida para produzir um dendrograma responsável por definir a ordem em que as sequências serão alinhadas. O fluxograma do funcionamento do T-Coffee é apresentado na Figura 3.4.

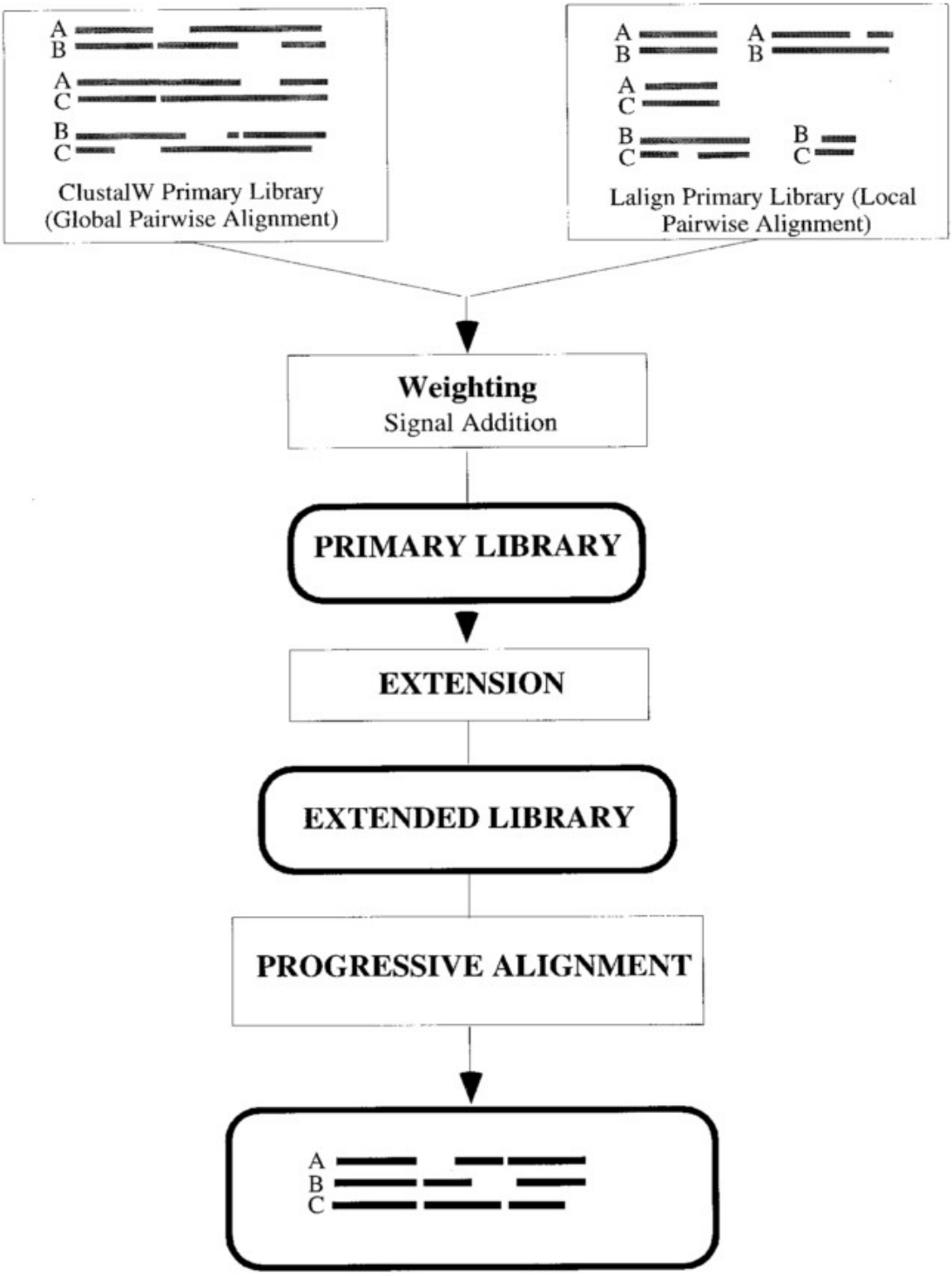

Figura 3.4: Fluxograma geral do T-Coffee
fonte: (NOTREDAME; HIGGINS; HERINGA, 2000)

Uma avaliação de Lassman e Sonnhammer (LASSMANN; SONNHAMMER, 2002) apontou o T-Coffee como um método mais confiável que o ClustalW para casos de sequências de baixa à média distância evolucionária. Contudo, o T-Coffee possui problemas em manipular muitas sequências de comprimento elevado (> 10.000) devido à sua alta demanda

computacional necessária (SIEVERS *et al.*, 2011).

3.3.5 SAGA

O SAGA (NOTREDAME, 1996) é um método de alinhamento de sequências iterativo estocástico que usa um algoritmo genético para selecionar o melhor alinhamento dentre uma população de alinhamentos que evoluem durante iterações. A função objetivo usada na seleção dos alinhamentos é a s*um of pairs* que é responsável por determinar o grau de aptidão dos indivíduos. Esta função define um esquema de pontuação onde, para cada coluna é calculado o somatório do grau de correspondências entre as bases o qual é identificado através de matrizes de similaridades. Nestas matrizes é definido o grau de similaridade entre todos os pares de bases possíveis. A Figura 3.5 apresenta uma matriz de similaridade BLOSUM62 (HENIKOFF; HENIKOFF, 1992) que é amplamente utilizada na literatura para alinhamentos múltiplos de sequências de proteínas. Quando aplicada a sequências de DNA, a função *sum-of-pairs*, faz uso de matrizes mais simples como a matriz IUB na qual a pontuação de qualquer correspondência exata vale 1,9 enquanto que as não-correspondências valem zero.

```
C  9
S -1  4
T -1  1  5
P -3 -1 -1  7
A  0  1  0 -1  4
G -3  0 -2 -2  0  6
N -3  1  0 -2 -2  0  6
D -3  0 -1 -1 -2 -1  1  6
E -4  0 -1 -1 -1 -2  0  2  5
Q -3  0 -1 -1 -1 -2  0  0  2  5
H -3 -1 -2 -2 -2 -2  1 -1  0  0  8
R -3 -1 -1 -2 -1 -2  0 -2  0  1  0  5
K -3  0 -1 -1 -1 -2  0 -1  1  1 -1  2  5
M -1 -1 -1 -2 -1 -3 -2 -3 -2  0 -2 -1 -1  5
I -1 -2 -1 -3 -1 -4 -3 -3 -3 -3 -3 -3 -3  1  4
L -1 -2 -1 -3 -1 -4 -3 -4 -3 -2 -3 -2 -2  2  2  4
V -1 -2  0 -2  0 -3 -3 -3 -2 -2 -3 -3 -2  1  3  1  4
F -2 -2 -2 -4 -2 -3 -3 -3 -3 -3 -1 -3 -3  0  0  0 -1  6
Y -2 -2 -2 -3 -2 -3 -2 -3 -2 -1  2 -2 -2 -1 -1 -1 -1  3  7
W -2 -3 -2 -4 -3 -2 -4 -4 -3 -2 -2 -3 -3 -1 -3 -2 -3  1  2 11
   C  S  T  P  A  G  N  D  E  Q  H  R  K  M  I  L  V  F  Y  W
```

Figura 3.5: Matriz de substituição BLOSUM62

Na matriz BLOSUM62, pares de aminoácidos mais raros recebem pontuação mais elevada.

Além disso, o SAGA conta com um grande número de operadores de variação e um mecanismo de seleção de quais operadores são utilizados em cada mutação ou cruzamento de acordo com o sucesso de cada operador em promover melhorias. A Figura 3.6 ilustra um dos operadores de cruzamento utilizado no método. Como se trata de um método iterativo e estocástico, possui a capacidade de encontrar várias soluções ótimas (caso existam). Outra característica do SAGA é a independência entre as definições da heurística de busca e a função objetivo, o que o torna mais flexível. Devido à sua estratégia de busca, possui a capacidade de semear a população inicial com boas soluções já conhecidas que podem ser obtidas por outros métodos como o ClustalW.

O SAGA provou ser capaz de produzir alinhamentos tão bons ou até melhores que o MSA e o ClustalW. Porém, apesar de encontrar alinhamentos ótimos para conjuntos de 30 sequências, o tempo de processamento foi extremamente alto (na ordem de milhares de vezes.

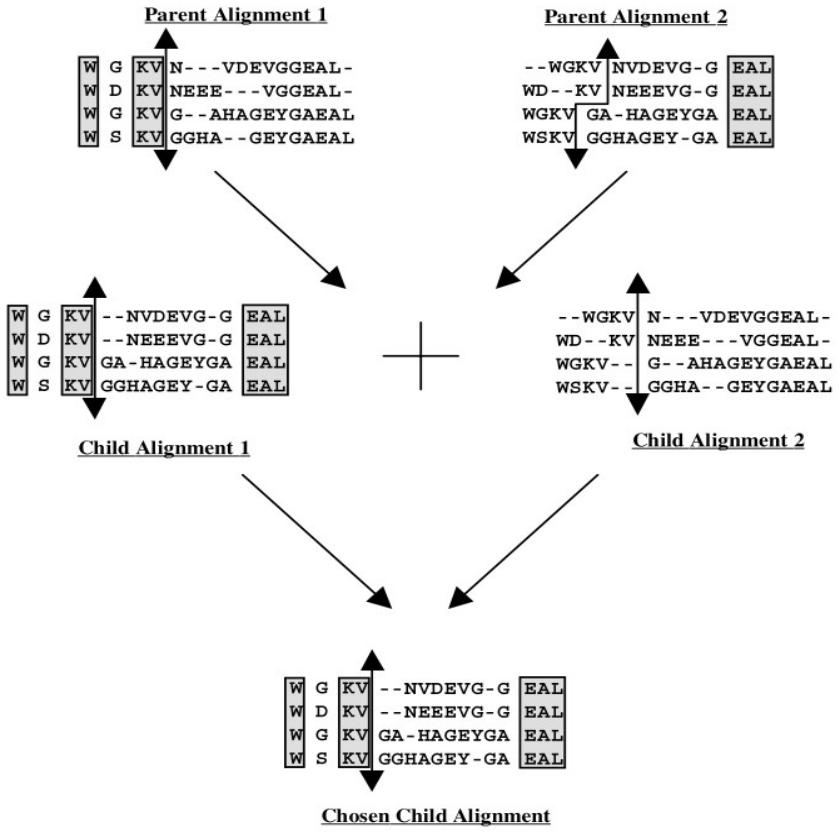

Figura 3.6: Operador de cruzamento produzindo novo alinhamento
fonte:(NOTREDAME, 2002)

4 PRINCÍPIOS DE COMPUTAÇÃO EVOLUCIONÁRIA

A Computação Evolucionária (CE) é uma das principais metodologias da Inteligência Computacional. O principal objetivo da Computação Evolucionária é prover ferramentas para a construção de sistemas inteligentes capazes de modelar comportamento inteligente (EBERHART; SHI, 2007). Para isso, a Computação Evolucionária define suas estratégias e formalismos baseando-se em elementos provenientes da Genética, da Teoria da Evolução de Darwin (FERREIRA, 1990) e de paradigmas baseados em comportamento emergente adaptativo (EBERHART; SHI, 2007). Desenvolvido recentemente (2009), o método Dialético de Otimização (DOS SANTOS; DE ASSIS, 2009), o qual baseia-se conceitos da Filosofia da Praxis, destaca-se por sua capacidade de otimização de diversas classes de funções (DOS SANTOS; DE ASSIS, 2009).

O campo da Computação Evolucionária pode ser dividido em basicamente cinco famílias de paradigmas (EBERHART; SHI, 2007): 1. Algoritmos genéticos; 2. Programação evolucionária; 3. Estratégias evolutivas; 4. Programação genética; 5. Otimização por enxame de partículas. Dessas, serão tratadas neste trabalho apenas as seguintes abordagens: algoritmos genéticos, programação evolucionária e otimização por enxame de partículas, pois formam a base principal de inspiração para o formalismo do método dialético objetivo como método de busca e otimização heurística, além da própria dialética materialista (DOS SANTOS, 2009).

4.1 VISÃO GERAL DE UM ALGORITMO EVOLUCIONÁRIO

As diversas variantes ou classes de algoritmos evolucionários (AEs) são baseadas na teoria da evolução. Assim, os AEs são compostos de 3 elementos básicos: (i) uma população de indivíduos os quais representam possíveis soluções para o problema em questão, (ii) uma função objetivo capaz de avaliar o quão bom (apto) é um indivíduo para a solução deste problema e (iii) um processo iterativo que, em cada iteração, promove alterações nos indivíduos e, posteriormente, seleção de parte deles para compor a população na próxima iteração.

Assim, os AEs compartilham características em comum: (1) são baseados em população, (2) usam operadores de variação na produção de novos indivíduos para a população, (3) fazem uso de uma função objetivo para medida da aptidão de cada indivíduo, (4) são estocásticos e (5) exigem a definição de uma representação capaz de modelar candidatos à solução como estruturas de dados codificadas que são manipuladas pelo algoritmo.

A definição formal de um AE inicia-se pela definição dos dois componentes principais que são dependentes do problema: (1) a representação no espaço de busca dos candidatos à solução e (2) a função de qualidade a ser maximizada. Segue-se então com a definição dos demais componentes do AE dentre os quais destacam-se a população, o mecanismo de seleção de pais e de sobreviventes, os operadores de Variação e os procedimentos de inicialização e de parada do algoritmo.

A execução de um AE parte da inicialização da população com candidatos à solução. Em seguida, a aptidão dos candidatos é calculada através da aplicação da função de objetivo. Alguns dos melhores candidatos são selecionados para produzir novos candidatos através da aplicação dos operadores estocásticos de variação. Então, alguns candidatos à solução são selecionados segundo alguma heurística para compor a população da próxima geração. Este processo se repete até que um determinado número de iterações seja atingido ou que alguma outra condição de término seja satisfeita. O funcionamento geral de um algoritmo evolucionário pode ser visto na Figura 4.1.

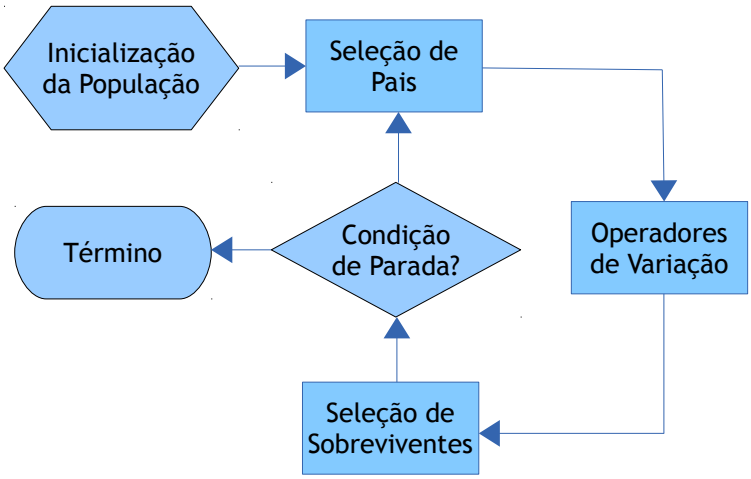

Figura 4.1: Esquema geral de um algoritmo evolucionário

Neste contexto, a busca pela solução ótima pode ser vista como uma otimização (ou aproximação) da função de qualidade pois é esperado que a aptidão cresça devido a sucessivas seleções ao longo de várias gerações.

Nas subseções seguintes, os principais componentes e procedimentos de um algoritmo evolucionário serão descritos.

4.1.1 Representação

Em Computação Evolucionária, elementos que formam possíveis soluções dentro do contexto original do problema são frequentemente chamados de fenótipos enquanto que suas codificações, ou seja, os indivíduos dentro do universo de busca do AE, são chamados de genótipos. O mapeamento entre fenótipo e um conjunto de genótipos é chamado de representação. Assim, é na definição da representação que as características relevantes de um possível candidato à solução são codificadas em uma estrutura de dados (genótipo). Segundo Eiben e Smith (2007), a representação pode ser vista como "o mapeamento entre o mundo real e o mundo do EA, ou seja, a ponte entre o contexto original do problema e espaço de solução do problema no qual a evolução ocorre".

4.1.2 Função de Objetivo

Também comumente chamada de função de aptidão, o papel da função objetivo é representar os requisitos que a evolução da população deve satisfazer. Na prática, é uma função matemática que atribui uma medida de qualidade aos candidatos à solução e sua definição é uma das tarefas mais importantes e complexas na aplicação de um algoritmo evolucionário para resolver um problema.

4.1.3 População

A população é um multiconjunto[6] de genótipos que representam possíveis soluções para o

[6] Multiconjunto é um conjunto onde múltiplas cópias de um objeto são possíveis.

problema em questão. Segundo a teoria da evolução, é a população que forma a unidade de evolução: os indivíduos são, na verdade, objetos estáticos e não mudam ou se adaptam – quem o faz é a população.

Definir uma população requer basicamente estabelecer seu tamanho. Existem AEs onde a população possui uma estrutura espacial adicional, definida via uma relação de vizinhança (DOS SANTOS; DE ASSIS, 2009; KENNEDY; EBERHART, 1995). Normalmente, a população possui tamanho fixo, entretanto existem paradigmas onde o tamanho da população varia ao longo da evolução (DOS SANTOS; DE ASSIS, 2009; DOS SANTOS, 2009).

A diversidade é um importante conceito associado à população e corresponde ao grau de diferença (distância) entre seus indivíduos (genótipos). A medição de diversidade é comumente definida por meio de medidas estatísticas como a entropia. A manutenção da diversidade é considerada uma boa estratégia para evitar a convergência prematura a ótimos locais (BACK; FOGEL; MICHALEWICZ, 1997; URSEM, 2002). Na definição de paradigmas mais recentes (DOS SANTOS; DE ASSIS, 2009) o controle da diversidade da população é implícito.

4.1.4 Mecanismo de Seleção de Pais

O papel deste mecanismo é permitir que os melhores indivíduos da população participem da criação dos indivíduos da próxima geração. Isso é feito basicamente selecionando os candidatos a pais baseando-se em suas qualidades (mensuradas através da função de aptidão). Um indivíduo é considerado um pai caso a ele sejam aplicados operadores de variação que produzam novos indivíduos.

Em Computação Evolucionária, a seleção de pais é normalmente probabilística, ou seja, indivíduos de qualidade elevada possuem mais chances de serem escolhidos quando comparados aos de mais baixa qualidade, porém ainda assim estes últimos recebem uma pequena, mas positiva, chance de serem escolhidos. Isso é justificado para ajudar na manutenção de diversidade fazendo assim que a busca por soluções não se torne tão ávida e gananciosa e acabe presa em um ótimo

local (BÄCK, 1996; EIBEN; SMITH, 2007).

4.1.5 Operadores de Variação

O papel dos operadores de variação é criação de novos indivíduos baseando-se nas características de indivíduos já existentes. Altamente dependentes da representação adotada, os operadores de variação são normalmente divididos em dois tipos de acordo com suas aridades[7].

4.1.5.1 Mutação

Um operador unário[8] de variação é comumente chamado de mutação. Ele é aplicado a um genótipo e produz sua cópia levemente modificada a qual é chamada de prole ou filho. Devido à sua natureza estocástica, é capaz tanto de intensificar buscas em regiões promissoras já conhecidas (explotação ou *explotation*) como também de visitar regiões ainda não conhecidas do espaço de busca (exploração ou exploration).

4.1.5.2 Recombinação ou Cruzamento

Recombinação ou cruzamento (*crossover)* é um operador de variação, geralmente binário[9], o qual combina as informações de dois genótipos pais para produzir um ou dois genótipos filhos. Assim como o operador de mutação, também é estocástico, portanto as escolhas de quais partes de cada pai serão recombinadas dependem de sorteios aleatórios.

O princípio do operador de cruzamento é simples: a combinação de dois indivíduos com diferentes, porém desejáveis características pode produzir novos indivíduos que possuam as qualidades de ambos. Em CE, a aplicação dos operadores de recombinação é usualmente probabilística, ou seja, existe uma chance diferente de zero de que o operador não seja de fato executado.

4.1.6 Mecanismo de Seleção por Sobrevivência (Substituição)

O papel da seleção de sobrevivência é, assim como na seleção de pais, fazer uma distinção

[7] Aridade de um operador é a quantidade de argumentos ou operandos recebidos na entrada
[8] Um operador é unário caso ele receba um único argumento ou operando em sua entrada
[9] Um operador é binário caso ele receba dois argumentos ou operandos em sua entrada

entre os indivíduos de melhor qualidade, porém em um diferente estágio do ciclo de evolução: após a produção de filhos através da aplicação dos operadores de variação. Neste momento a população tem seu tamanho expandido consideravelmente e devido a limites computacionais óbvios, a quantidade de indivíduos na população deve ser limitada. Então, cabe à seleção por sobrevivência identificar quais indivíduos serão permitidos permanecerem na população para a próxima geração. Diferente da seleção de pais, a qual é tipicamente estocástica, a seleção por sobrevivência segue frequentemente um modelo determinístico.

4.1.7 Inicialização

O processo de inicialização da população é normalmente definido de maneira simples nos AEs: a população inicial é semeada com indivíduos produzidos aleatoriamente. Heurísticas específicas aos problemas (BRAMLETTE, 1991) podem ser usadas para a criação de uma população inicial de aptidão mais alta.

4.1.8 Condição de Término

O processo de terminação pode ser definido levando-se em conta o problema em questão. Isso é permitido quando um nível de aptidão ótimo é conhecido pois, em condições ideais o algoritmo deveria ser executado até que tal valor (ou valor próximo) fosse encontrado. Porém, como os algoritmos evolucionários são estocásticos, na maioria dos casos não há garantias de alcançar tal valor ótimo e, assim, o algoritmo evolucionário pode nunca terminar. Então, a condição de parada é definida sob a composição de outras condições que possuam garantias de término dentre as quais são mais utilizadas o (1) tempo de uso máximo de CPU é esgotado, (2) o total de avaliações feitas pela função de aptidão alcança um limiar, (3) a melhoria de aptidão permanece abaixo de dado um limite durante um período de tempo ou quantidade de gerações estipulados e (4) a diversidade da população cai abaixo de um dado limiar.

4.2 ALGORITMOS GENÉTICOS

Os Algoritmo Genéticos (*Genetic Algorithms*, GAs) compreendem uma família de modelos computacionais que definem uma heurística de busca que imita o processo de seleção natural inspirando-se na Teoria da Evolução de Darwin e na genética de Mendel. Inicialmente, o GA foi concebido como um método de estudo ao comportamento adaptativo (HOLLAND, J., 1992). Entretanto, a literatura mostra que o GA tem sido largamente utilizado como método de otimização de funções, aprendizado de máquina e outras classes de problemas (EIBEN; SMITH, 2007; MITCHELL, 1996; WHITLEY, 1994).

O processo de busca realizado pelo GA segue o modelo geral dos algoritmos evolucionários exibido na Figura 2.1. A construção deste é iniciada a partir da definição dos dois únicos componentes principais dependentes do problema que são (1) a codificação do fenótipo (representação) e (2) a função de qualidade (ou aptidão) (WHITLEY, 1994).

4.2.1 Representação

Em sua forma original, conhecida por SGA ou *Simple Genetic Algorithm* (HOLLAND, J. H., 1973), o GA estabelece a representação do genótipo como cromossomos compostos de genes cuja única variação permitida é a de estarem ou não ativos. Assim, no universo de busca de um SGA, a estrutura de dados (genótipo) que representa um indivíduo é definida por um vetor de bits (genes) cujos valores (alelos) correspondem ao estado de ativação associado a cada característica (fenótipo) considerada na modelagem fenótipo – genótipo. No SGA a seleção de pais é probabilística e baseada na aptidão enquanto que a seleção por sobrevivência é baseada na forma mais simples de seleção por idade na qual nenhum indivíduo da geração atual segue para próxima geração.

Comumente são encontradas variações do SGA aplicadas a outros contextos como em otimização numérica, no aprendizado de máquina e outros. Nestas, foram modeladas outras

representações baseadas em vetores de números reais, vetores de números inteiros e vetores de permutação de valores de um alfabeto limitado.

4.2.2 Mecanismos para Seleção de pais

Nos GAs, o esquema estocástico de seleção de pais possui duas formas de obtenção da probabilidade de escolha de cada indivíduo: um baseado em aptidão absoluta ou outro em aptidão relativa (posição).

4.2.2.1 Probabilidade proporcional à aptidão absoluta

Neste caso, a seleção probabilística é realizada proporcionalmente ao valor absoluto de aptidão de cada indivíduo. Assim, cada indivíduo *i* tem uma chance $f_i / \sum_{i=1}^{\mu} f_j$ de ser escolhido como pai, onde f_i corresponde ao valor da função de aptidão *f* aplicada ao *j*-ésimo indivíduo, com $1 \leq j \leq \mu$. Tal mecanismo de seleção é reconhecido por possuir problemas como:

1. Rápida redução da diversidade pois indivíduos de aptidão muito elevada tomam toda população rapidamente levando à convergência prematura.
2. A redução da pressão de seleção em casos com uma população com pequena variação de aptidão entre os indivíduos onde as chances de ser escolhido são praticamente iguais e ter ou não uma aptidão não é tão útil a um indivíduo.
3. O comportamento desse mecanismo comporta-se diferentemente em versões transpostas de uma mesma função de aptidão como pode ser visto na Figura 4.2 (EIBEN; SMITH, 2007).

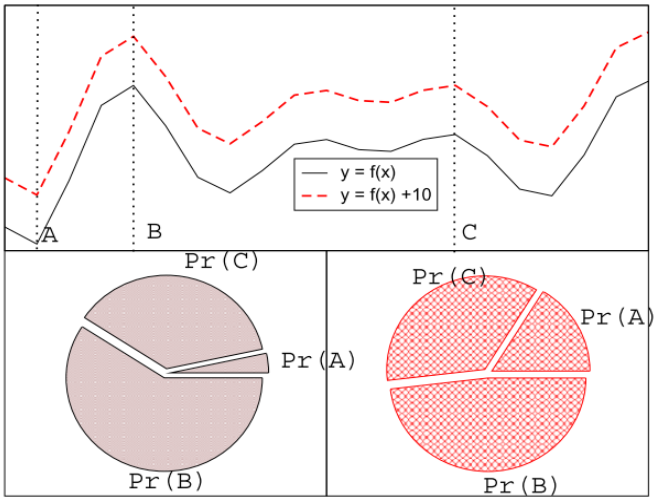

Figura 4.2: Alteração de comportamento no cálculo de probabilidade de seleção.

Acima: Nível de aptidão em três pontos (A,B e C) para *f(x)* e *f'(x)=f(x)+10*
Abaixo à esq.: Probabilidade de seleção para os três pontos sobre *y = f(x)*
Abaixo à dir.: Probabilidade de seleção para os três pontos sobre *y = f'(x)*

4.2.2.2 Probabilidade proporcional à posição relativa em lista ordenada pela aptidão

Nesta abordagem, os indivíduos são ordenados segundo sua aptidão, e a seleção de um indivíduo é feita baseado-se na posição ocupada por ele. Dessa forma, a escolha desse tipo de seleção mantém a pressão de seleção constante como também evita a convergência prematura e a comportamento não esperado quando da transposição da função de aptidão.

A escolha da expressão matemática que define como a probabilidade de escolha varia em função da posição varia conforme o nível de pressão de seleção desejado.

4.2.2.3 Seleção dos Pais

De posse das probabilidades de cada indivíduo, a seleção dos pais é feita através de sorteio por uma roleta. Esse método é equivalente ao uso de uma roleta tradicional de cassino onde a cada indivíduo é reservada uma área do disco proporcional à sua probabilidade de escolha. O pseudocódigo do algoritmo para seleção por disco de roleta é ilustrado na Figura 4.3.

```
BEGIN
  set membro_atual = 1;
  WHILE (membro_atual <= μ) DO
    Tomar um valor aleatorio r uniformemente de [0,1];
    set i = 1;
    WHILE ( a_i < r ) DO
      set i = i + 1;
    set futuros_pais[membro_atual] = pais[i];
    set membro_atual = membro_atual + 1;
END
```

Figura 4.3: Pseudocódigo para algoritmo de seleção por roleta
fonte: adaptado de (EIBEN; SMITH, 2007)

Porém, quando a população é extensa, é comum utilizar outro método chamado seleção por torneio. Nele, cada membro do conjunto dos pais é escolhido tomando-se o mais apto dentre uma amostra de *k* indivíduos da população escolhidos ao acaso. O pseudocódigo para algoritmo de seleção por torneio de μ pais pode ser visto na Figura 4.4.

```
BEGIN
  set membro_atual = 1;
  WHILE (membro_atual <= μ) DO
    Tomar k indivíduos aleatoriamente, ou ou sem reposição;
    Selecionar o melhor destes k comparando suas aptidões;
    Chamar este melhor indivíduo de i;
    set matting_pool[membro_atual] = i;
    set membro_atual = membro_atual + 1;
END
```

Figura 4.4: Pseudocódigo para algoritmo de seleção por torneio de μ pais
fonte: adaptado de (EIBEN; SMITH, 2007)

4.2.3 Operadores de Variação

4.2.3.1 Mutação

É um operador de variação que toma um único pai e cria um único filho aplicando algum tipo de mudança aleatória sobre a representação (genótipo) do pai. O parâmetro associada ao operador de mutação é comumente chamado de taxa ou passo de mutação. O ajuste desse parâmetro normalmente segue padrões definidos experimentalmente (DE JONG, 1975) embora, em determinados problemas seja necessário um ajuste mais específico. Assim como qualquer operador de variação, sua definição dependem da representação utilizada (EIBEN; SMITH, 2007).

Para as representações inteira e real, os operadores de mutação mais comumente usados são *Random Resetting*, o qual sorteia segundo uma distribuição uniforme um novo valor para o gene, e *Creep Mutation*, onde é um pequeno valor é subtraído ou adicionado ao do gene. O primeiro é mais adequado à explotação de novas regiões promissoras enquanto que o último se concentra mais na exploração em áreas já conhecidas.

4.2.3.2 Recombinação

O operador de recombinação pode ser considerada um das características que mais distingue o GA, como também outros algoritmos evolucionários que usam recombinação, de outros algoritmos de otimização global. Este é um operador é capaz de criar um ou mais indivíduos através da combinação de dois ou mais indivíduos já existentes e é aplicado probabilisticamente de acordo com o parâmetro conhecido por taxa de recombinação ou probabilidade de cruzamento cujo valor típico repousa entre [0,5; 1,0].

Na aplicação de um operador de cruzamento, usualmente dois pais são selecionados e um número aleatório é sorteado uniformemente do intervalo [0,0; 1,0) e comparado à probabilidade de cruzamento. Caso o número sorteado seja menor que probabilidade de cruzamento, dois filhos são produzidos através da recombinação do genótipo dos pais. Caso contrário, os filhos produzidos são cópias idênticas aos pais. Assim, em contraste com o operador de mutação, no qual a taxa de mutação determina se o operador será ou não aplicado a um gene específico, no operador de cruzamento a taxa de cruzamento determina se o operador será ou não aplicado aos indivíduos pais envolvidos.

Para representações binária, inteiras e reais os operadores de cruzamento mais utilizados são *one-point*, *n-point* e *uniform crossover*. No operador *one-point*, um ponto de corte (posição) é escolhida ao acaso. O novo indivíduo é produzido tomando os genes do primeiro Pai localizados à esquerda do ponto de corte e concatenando-os com os genes do segundo Pai localizados à direita do ponto de corte conforme a Figura 4.5 (EIBEN; SMITH, 2007). Um segundo filho pode ser produzido mantendo-se o ponto de corte e invertendo-se os Pais.

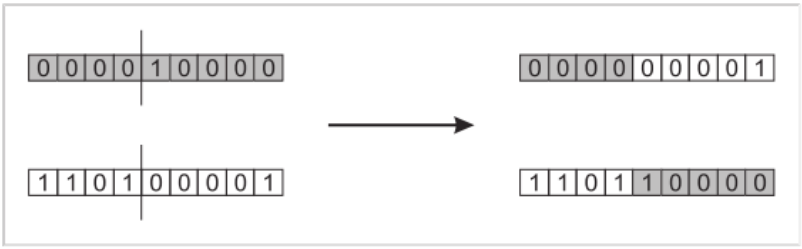

Figura 4.5: Operador one-point crossover

O operador *n-point* funciona de forma equivalente, porém são escolhidos vários pontos de corte (produzindo assim vários seguimentos) e os segmentos são tomados de cada pai de forma intercalada Figura 4.6 (EIBEN; SMITH, 2007).

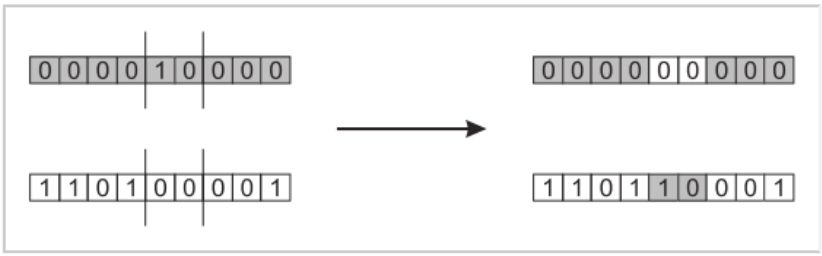

Figura 4.6: Operador n-point crossover

No *uniform crossover,* para cada posição é sorteado uniformemente um número de [0,0; 1,0]. Se o número for menor que a probabilidade de cruzamento, então o gene da posição atual é herdado primeiro pai, caso contrário é herdado do segundo pai. Um segundo filho pode ser produzido utilizando o mapeamento inverso conforme a Figura 4.7 (EIBEN; SMITH, 2007).

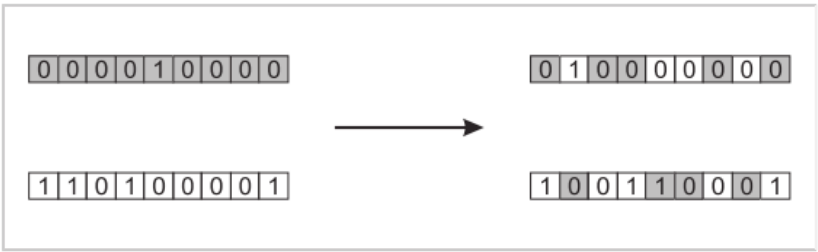

Figura 4.7: Operador uniform crossover

Para representações de vetores de números reais existe uma classe especial de operadores chamada de recombinação aritmética. Trata-se de uma versão de operadores já conhecidos na qual o gene do filho não recebe cruzados mas sim a média entre os valores herdados de ambos os pais de acordo com a Figura 4.8 (EIBEN; SMITH, 2007).

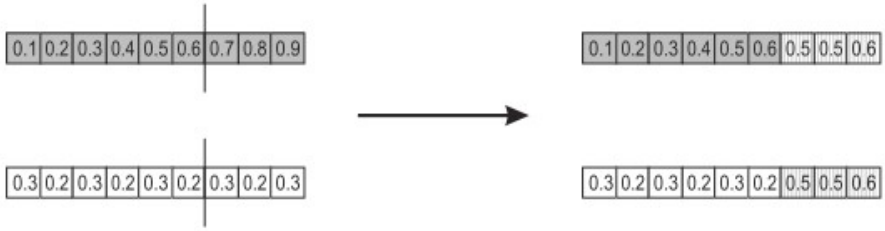

Figura 4.8: Exemplo de operador aritmético de cruzamento

4.2.4 Seleção por sobrevivência

Os diversos mecanismos existentes de seleção de sobreviventes para GAs podem ser classificados em duas categorias. A primeira categoria é composta daqueles mecanismos cuja substituição dos indivíduos é baseada em idade. Por exemplo, no SGA, todos indivíduos da geração atual são descartados após a geração de filhos por meio dos operadores de variação. Nesse caso, é comum o uso de uma técnica chamada elitismo na qual indivíduos mais aptos da população são preservados para a próxima geração. Na outra categoria, a substituição de indivíduos é baseada na aptidão na qual diversos esquemas de substituição podem ser usados inclusive equivalentes àqueles usados na seleção de pais como seleção proporcional a aptidão e por torneios, porém aplicando-os à aptidão inversa.

4.2.5 Comentários

Algoritmos Genéticos podem ser usados com sucesso na solução de problemas diversos problemas de otimização como problemas de controle de processo, classificação e reconhecimento de padrões (BRAGA; CARVALHO; LUDEMIR, 2000). Também é possível solucionar problemas de otimização multiobjetivos através de algoritmos genéticos (BINGUL; SEKMEN; ZEIN-SABATTO, 1923), porém a modelagem dos candidatos a solução na forma de cromossomos é uma tarefa crítica (BRAGA; CARVALHO; LUDEMIR, 2000; EBERHART; SHI, 2007). Outro ponto relevante é que embora o GA possa ajudar na busca da solução ótima ou sub-ótima, sua aplicação demanda um custo computacional relativamente alto. Além disso, o GA também é muito sensível à

modelagem dos fenótipos em genótipos (BRAGA; CARVALHO; LUDEMIR, 2000; EBERHART; SHI, 2007).

4.3 PROGRAMAÇÃO EVOLUCIONÁRIA

A Programação Evolucionária ou EP (*Evolutionary Programming*), outra importante família de técnicas de Computação Evolucionária, foi originalmente concebida por Lawrence J. Fogel em 1966 (FOGEL, L. J.; OWENS; WALSH, 1966) para empregar evolução simulada como uma abordagem de Inteligência Artificial. Porém, desde os anos de 1990, variações de EP para otimização de vetores de parâmetros reais passaram a se tornar mais frequentes sendo posicionados como o EP padrão (BÄCK; SCHWEFEL, 1993; BÄCK, 1996).

Também de natureza estocástica, a EP difere-se do GA principalmente por (1) não utilizar representação na forma de cromossomos e (2) por não realizar seleção de pais e operações de recombinação. O fluxograma geral do EP pode ser visto na Figura 4.9.

4.3.1 Representação

Nos EPs, os indivíduos são representados por vetores de números reais e normalmente o parâmetro taxa de mutação é inserido ao genótipo. Isso ocorre para que o controle da frequência das mutações seja direcionado pelo próprio processo evolutivo (a tal situação dá-se o nome meta-evolução). A importância disso cai sobre o fato que diferentes tamanhos de passos de mutação são necessários para enfatizar a descoberta de novas regiões promissores (exploração), ocorrida geralmente no início da busca, e a explotação de boas regiões já identificadas a qual é fundamental para convergência ao ótimo global (FOGEL, D. B., 1995).

4.3.2 Seleção de pais e Seleção de Sobrevivência

No EP não ocorre seleção de pais e todos os μ indivíduos da população produzem exatamente um filho através do operador de mutação. Portanto, ao final deste processo, haverão

μ+μ indivíduos na população. Já a seleção de sobreviventes é realizada de forma estocástica onde, tipicamente, um abordagem baseada em competição por torneios em pares é utilizada.

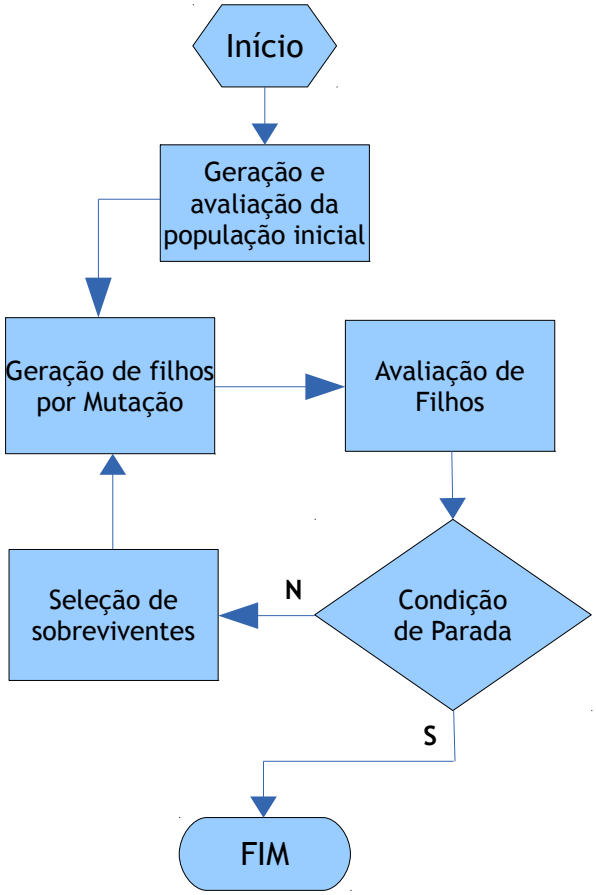

Figura 4.9: Fluxograma geral do EP

Nela cada indivíduo *i* é comparado com outros *q* indivíduos (*q* = 10 é recomendado) tomados aleatoriamente da população. Em cada comparação, uma vitória é atribuída caso o indivíduo *i* seja melhor que seu oponente. No final, os μ indivíduos mais vitoriosos serão mantidos para a próxima geração. A pressão que a aptidão exerce na seleção pode ser alterada através do aumento do parâmetro *q* pois, quanto mais adversários houverem, maior a chance de um indivíduos de baixa aptidão sortear oponentes mais aptos durante o torneio reduzindo assim suas chances de sobrevivência.

4.3.3 Mutação

A mutação transforma um indivíduo da forma $\langle x_1, x_2, \ldots, x_n, \sigma_1, \sigma_2, \ldots \sigma_n \rangle$ em outro da

forma $\langle x'_1, x'_2, \ldots, x'_n, \sigma'_1, \sigma'_2, \ldots \sigma'_n \rangle$ onde *n* corresponde à dimensão do problema e

$$\sigma'_i = \sigma_i (1 + \alpha . X),$$
$$x'_i = x_i + \sigma'_i + X_i .$$

onde $X(0,1)$ corresponde a um número aleatoriamente tomado de uma distribuição de probabilidade específica e $\alpha \approx 0{,}2$ (EIBEN; SMITH, 2007). Na definição clássica de EP, conhecida como CEP, *Classical Evolutionary Programming*, a função de distribuição utilizada é a função de distribuição gaussiana com média zero e desvio padrão igual a um. O algoritmo CEP funciona da seguinte maneira (SHEN; HE, 2010; YAO; LIU; LIN, 1999):

1. Inicialmente é realizada a geração de uma população inicial de *m* indivíduos *n*-dimensionais, representados pelo par $(\mathbf{x}_i, \mathbf{s}_i)$, $\forall i \in \{1, 2, \ldots, m\}$, onde $\mathbf{x}_i$ são os valores os quais otimizam a função objetivo $f : S \rightarrow \mathbb{R}$, onde $S \subseteq \mathbb{R}^n$ e *n* é a dimensionalidade do problema, enquanto $\mathbf{s}_i = (\mathbf{s}_{i,1}, \mathbf{s}_{i,2}, \ldots, \mathbf{s}_{i,n})^T$ são os desvios padrões para as mutações gaussianas, também chamados de parâmetros estratégicos em algoritmos evolucionários autoadaptativos;
2. Repetir até que um determinado critério de parada seja satisfeito:
   a) Avaliação da função objetivo *f* nos indivíduos pais, $(\mathbf{x}_i, \mathbf{s}_i)$, $\forall i \in \{1, 2, \ldots, m\}$, gerando $f(\mathbf{x}_i)$;
   b) Geração de *m* indivíduos filhos por mutação, $(\mathbf{x}'_i, \mathbf{s}'_i)$, $\forall i \in \{1, 2, \ldots, m\}$, de forma que:
   $$s'_{i,j} = s_{i,j} \exp(\tau_a G(0,1) + \tau_b G(0,1)),$$
   $$x'_{i,j} = x_{i,j} + s'_{i,j} G_j(0,1)$$
   $\forall i \in \{1, 2, \ldots, m\}$ e $\forall j \in \{1, 2, \ldots, m\}$, onde $G(0,1)$ é uma variável aleatória gaussiana de média 0 e variância 1 gerada para um *i* fixo enquanto que $G_j(0,1)$ é gerada para cada *j*; já os parâmetros de controle são definidos como $\tau_a = 1/\sqrt{2n}$ e $\tau_b = 1/\sqrt{2\sqrt{n}}$;
   c) Avaliação da função objetivo nos indivíduos filhos, $(\mathbf{x}'_i, \mathbf{s}'_i)$, $\forall i \in \{1, 2, \ldots, m\}$, gerando $f(\mathbf{x}'_i)$;
   d) Seleção dos *m* indivíduos mais aptos dentre a população de *2 m* indivíduos composta da junção das populações de pais e de filhos, para compor a nova geração de pais.

4.3.4 Outras abordagens

Na literatura existem versões estendidas do CEP que buscam melhorar seu desempenho de forma geral ou a problemas específicos. O algoritmo FEP, Fast Evolutionary Programming, proposto por Yao *et.al.* (YAO; LIU; LIN, 1999) faz uso da mutação Cauchy (SHEN; HE, 2010; YAO; LIU; LIN, 1999). O objetivo de é usar o formato da curva de Cauchy (ver Figura 4.10) para aumentar frequência mutações pequenas melhorando, assim, a chance de convergência para o ótimo global.

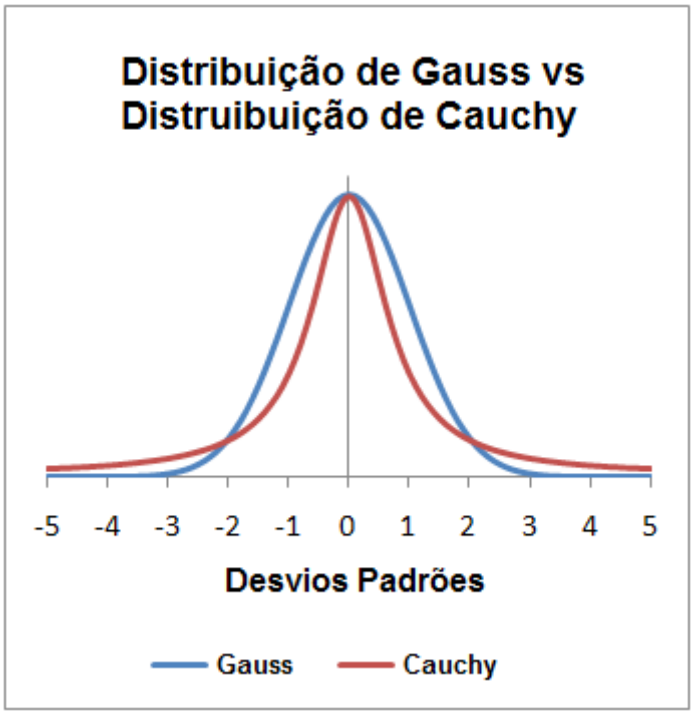

Figura 4.10: Comparativo entre as dispersões das curvas de distribuição: Curva de Gauss, em azul, e curva de Cauchy, em vermelho
fonte:(STACKEXCHANGE, 2012)

Outro algoritmo chamado LEP, *Lévy-Type* Evolutionary Programming, proposto por Iwamatsu (IWAMATSU, 2002), faz uso de uma mutação baseada na distribuição de probabilidade de Lévy a qual produz uma variável aleatória capaz de ser controlada através do parâmetro de escala β. O ajuste de β é capaz de controlar a eficiência do LEP para determinados tipos de funções. O LEP tem eficiência equivalente ao CEP quando $\beta = 2$ e ao FEP quando $\beta = 1$. Entretanto, a escolha do melhor valor de β para cada tipo de problema em específico é difícil.

4.3.5 Conclusões

Várias versões do EP foram sugeridas nos últimos quinze anos, porém a maioria delas baseadas na estratégia de mutação. A mutação gaussiana, usada no método clássico (CEP), converge bem com funções unimodais porém tem problemas na otimização de funções multimodais (SHEN; HE, 2010; YAO; LIU; LIN, 1999). Utilizada no FEP, a mutação Cauchy foi proposta como alternativa à mutação gaussiana e oferece melhores respostas à funções multimodais, porém o FEP é menos eficiente que o CEP em funções unimodais. Apresentada no algoritmo LEP, a mutação baseada na distribuição de Lévy, através do ajuste do parâmetro β, é capaz de funcionar bem com ambas classes de funções, porém é difícil fazer a melhor escolha de β para um determinado problema (SHEN; HE, 2010).

Os algoritmos EPs são relativamente simples de serem implementados, porém necessitam de muitas iterações até atingir bons resultados e também de uma população inicial relativamente grande e, consequentemente, em um número elevado de avaliações através da função objetivo, o que pode tornar sua aplicação inadequada para certas classes de funções .

4.4 Otimização por Enxame de Partículas

Otimização por enxame de partículas ou PSO (*Particle Swarm Optimization*) é um método para resolução de problemas de otimização criado em 1995 por James Kennedy, um sociólogo, e Russel Eberhart, um engenheiro eletricista (KENNEDY; EBERHART, 1995). É um algoritmo inspirado no comportamento social (no movimento coletivo) de bandos de animais como pássaros e peixes.

De forma semelhante ao GA, o PSO busca pela solução do problema de forma iterativa através da promoção de melhorias, a respeito de uma função de qualidade previamente definida, de candidatos à solução de uma população. No PSO, tal busca é baseada na movimentação desses candidatos à solução, chamados de partículas, sobre o espaço de busca. Enquanto que tais candidatos nos Algoritmos Genéticos são representados por cromossomos, no PSO cada candidato

está associado a um vetor posição e a outro vetor velocidade. Além disso, no PSO não ocorrem processos de seleção nem mutação de indivíduos: a cada iteração apenas as posições e velocidades das partículas são ajustadas em direção da melhor partícula local ou individual e da melhor partícula global. A melhor partícula global, $\boldsymbol{\rho}_g$, corresponde à partícula de maior qualidade relativo à função objetivo, ou seja:

$$f(\mathbf{p}_g) = \max_{1 \le j \le m} f(\mathbf{x}_j),$$

enquanto $\boldsymbol{\rho}_i$ significa melhor partícula individual ou local em relação à *i*-ésima partícula. Caso a implementação seja baseada em partícula individual, $\boldsymbol{\rho}_i$ corresponde a melhor posição visitada pela *i*-ésima partícula, ou seja:

$$f((\mathbf{x}_i(t''))) = \max_{0 \le t \le t'} f(\mathbf{x}_i(t')) \Rightarrow \mathbf{p}_i(t) = \mathbf{x}_i(t''),$$

nos casos em que a implementação seja baseada em partícula local, alguma função de vizinhança, a qual normalmente opera em função do índice *i,* define como N_i o conjunto de partículas vizinhas a *i*-ésima partícula. Assim, a melhor posição local é definida por

$$f(\mathbf{p}_i) = \max_{j \in N_i} f(\mathbf{x}_j),$$

e dessa forma, diferentes topologias podem ser montadas (ZAVALA; AGUIRRE; DIHARCE, 2005) para resolver problemas mais complexos, como os problemas de otimização multiobjetivo, baseando-se no conceito de coevolução onde cada grupo de enxames pode ser responsável em resolver um determinado objetivo.

De modo geral, o deslocamento da *i*-ésima partícula, na *t*-ésima iteração, é definido pela expressão canônica:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$

deste que

$$\mathbf{v}_i(t+1) = w\,\mathbf{v}_i(t) + c_1 r_1(t)(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 r_2(t)(\mathbf{p}_g(t) - \mathbf{x}_i(t)),$$

onde *m* é o número de partículas do enxame, com $1 \le i \le m$; *w* é o fator de inércia, onde $0 < w < 1$; $r_1(t)$ e $r_2(t)$ são números aleatórios uniformemente distribuídos no intervalo [0,1]; c_1 e c_2 são

constantes de constrição, também chamados de coeficientes de aceleração, onde $c_1+c_2=4$ (usualmente, $c_1 = 2 + \Delta$ e $c_2 = 2 - \Delta$, com $\Delta \approx 0$), sendo que c_1 é o peso devido à consciência individual da partícula (ou local), enquanto que c_2 é o peso devido à consciência global; $\mathbf{x}_i$ é a posição, enquanto $\mathbf{v}_i$ é a velocidade da *i*-ésima partícula; $\mathbf{p}_g$ é a melhor posição global.

Existem versões do POS (WANG *et al.*, 2006) que introduzem na expressão canônica melhorias produzidas pela introdução de conceitos físicos como velocidade de escape, resultando na seguinte expressão de ajuste de velocidades:

$$\mathbf{v}_i(t+1) = w\,\mathbf{v}_e(t) + c_1 r_1(t)(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 r_2(t)(\mathbf{p}_g(t) - \mathbf{x}_i(t)),$$

para

$$v_{e,j}(t) = \begin{cases} v_{i,j}(t), & |v_{i,j}(t)| > e_c(t) \\ \dfrac{1}{\rho} v_{max}\, r_j(t), & |v_{i,j}(t)| \le e_c(t) \end{cases},$$

onde $\mathbf{v}_e(t) = (v_{e,1}(t), v_{e,2}(t), \ldots, v_{e,n}(t),)^T$ é a *velocidade de escape*, para $1 \le i \le m$ e $1 \le j \le n$; v_{max} é a velocidade escalar máxima, assumindo tipicamente o maior valor da função objetivo; $r_j(t)$ é uma sequência de número aleatórios uniformemente distribuídos no intervalo [-1, 1] no instante *t*, ρ é o fator de escala para controlar o domínio da velocidade de escape; e $e_c(t)$ é o limiar ajustável que determina a condição de escape.

Por não possuir mecanismos de mutação e de seleção de indivíduos, o PSO é uma classe de algoritmos relativamente simples de serem implementados pois se resumem a um processo iterativo de atualização e avaliações dos candidatos à solução, conforme a Figura 4.11. Diversos resultados têm mostrado que o PSO pode ser aplicado com sucesso para uma classe relativamente grande de problemas, inclusive problemas multiobjetivos (normalmente resolvidos utilizando-se vários PSOs em cooperação) (EBERHART; SHI, 2007).

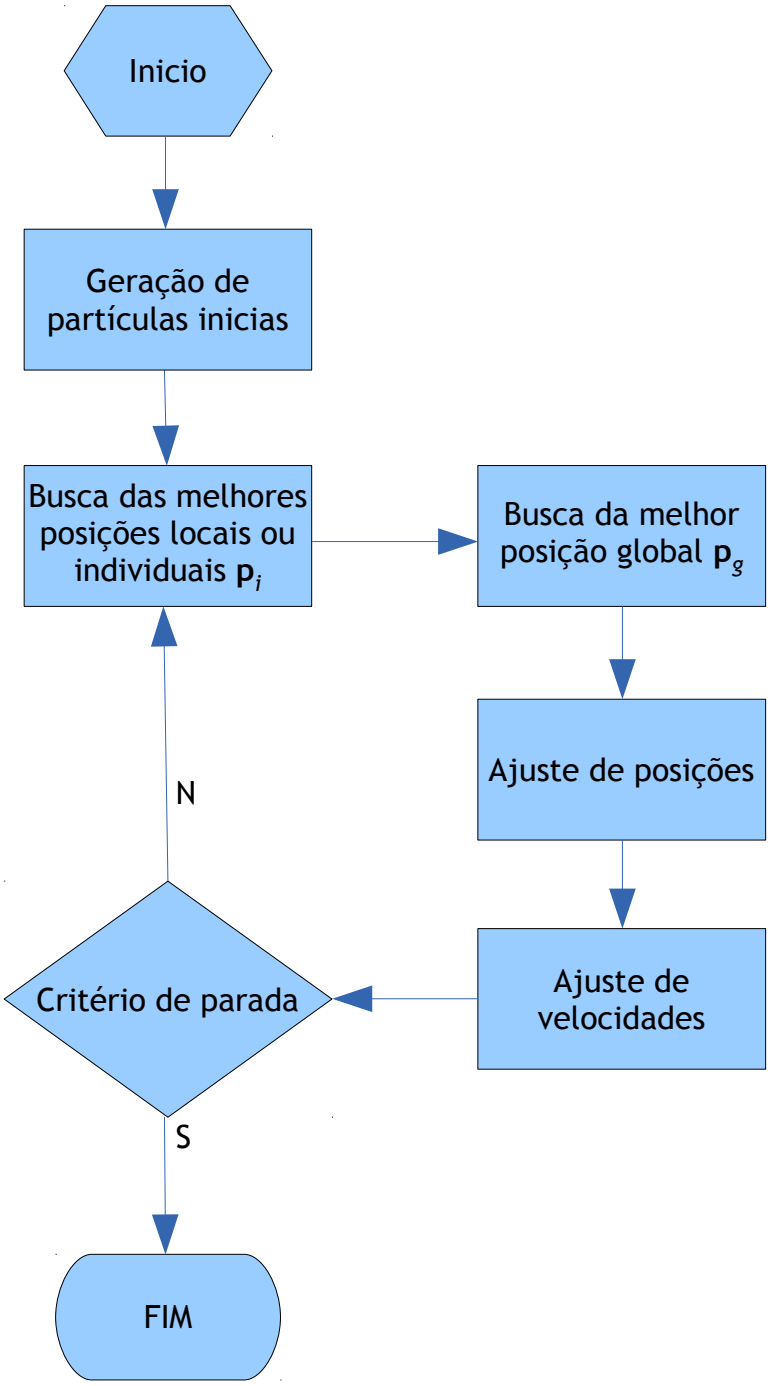

Figura 4.11: Fluxograma geral do PSO

5 OTIMIZAÇÃO DIALÉTICA

5.1 INTRODUÇÃO

O Método Dialético de Otimização ou *Optimization Dialectical Method*, ODM, é uma classe de algoritmo evolucionário baseada em uma interpretação específica da dialética materialista para solução de problemas de busca e otimização (DOS SANTOS; DE ASSIS, 2009; DOS SANTOS *et al.*, 2008).

A palavra dialética vem do original grego *dialetiké*, onde o prefixo *dia* equivale a interação, enquanto que *letiké* significa saber, palavra, conhecimento. Assim, dialética é vista como a livre interação entre várias ideias a respeito de um objeto específicos, sendo considerada capaz de obter informação sobre a verdade sobre tal objeto ou, ao menos, capaz de conseguir a vizinhança que envolve essa informação (VAZQUEZ, 2007).

A definição da base moderna do método dialético teve início no final do século 18 e início do século 19 através dos trabalhos do filósofo alemão George Wilhelm Friedrich Hegel, 1770-1831. Na época, Hegel procurava categorizar a natureza de pensamento dialético para aplicá-la em importantes questões filosóficas como a dinâmica da sociedade e as origens do Estado (MARX, 2002).

O ODM baseia-se na concepção de partes da realidade, ou fenômenos, como sistemas. Tais sistemas são compostos por diversos polos. Cada polo possui uma força. Assim, o sistema é caracterizado pelo balanceamento entre as forças dos polos que contém. (VAZQUEZ, 2007). Essa interação obedece ao modelo concebido por Hegel e é baseada em três conceitos fundamentais: a contradição, o princípio da totalidade e a movimentação sem fim (LÖWY, 2008). A contradição é a base da dialética e é definida pela correlação entre as forças dos polos ao longo do tempo. Assim, a dialética é o movimento das contradições.

Em (DOS SANTOS; DE ASSIS, 2009) a adaptação do método dialético para solução de problemas de otimização e busca é feita considerando cada polo como um candidato à solução do problema. Assim, a união de diversos polos corresponde ao universo de soluções do problema o qual define o espaço onde a busca pela solução ótima ocorrerá. A ideia principal do ODM, então, é associar a função objetivo do problema de otimização à força de cada polo, daí a dinâmica de otimização é determinada pela luta de polos. Nessa luta, o movimento dos polos ocorre em função do polo hegemônico presente e do polo hegemônico histórico. O polo hegemônico é aquele com a maior força social entre o conjunto das forças sociais de todos os polos em um determinado momento histórico. O polo hegemônico presente é o polo hegemônico do instante atual, enquanto o polo hegemônico histórico é o polo hegemônico do período histórico compreendido entre o início do sistema dialético até o instante atual.

A busca de soluções possíveis é intrinsecamente relacionada ao movimento dos polos e suas contradições nas várias fases históricas ao longo das quais estão incluídos períodos de crises revolucionárias, a fusão de polos com baixo nível de contradição entre si, a criação de novos polos através de sínteses ocorridas para resolver altas contradições e a geração de antíteses absolutas dos polos tese. Neste processo, também ocorrem pequenas pertubações aos polos sobreviventes ao período de crise: este é um processo um pouco similar à aplicação em massa de mutações e seu objetivo é incluir diversidade durante a busca.

5.2 Definição Geral

A definição formal do método dialético requer a definição dos seguintes conceitos (DOS SANTOS, 2009):

- Polo: É a unidade fundamental do sistema dialético. No contexto de método de otimização, o polo executa o papel de candidato à solução. Dado um conjunto de polos $\Omega=\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}$ um polo $\mathbf{w}_i$ está associado a um vetor de pesos

$\mathbf{w_i}=(w_{i,1}, w_{i,1}, \ldots, w_{i,n})^T$ onde $\mathbf{w}_i \in S$, m é o número de polos e n é a dimensionalidade do sistema.

- Força social, $f(\mathbf{w}_i)$: Corresponde ao valor da função objetivo f quando aplicada ao polo $\mathbf{w}_i$.
- Hegemonia: No processo de luta entre polos, um polo k exerce a hegemonia no instante t se $f(\mathbf{w}_k(t)) = f_C(t) = \max_{1 \le j \le m(t)} f(\mathbf{w}_j(t))$, onde $k \in \{1,2,\ldots, m(t)\}$. O vetor $\mathbf{w}_C(t) = \mathbf{w}_k(t)$ é chamado polo hegemônico presente ou polo hegemônico contemporâneo, enquanto que é a força hegemônica presente ou contemporânea. A força hegemônica histórica no instante t, $f_H(t)$, é dada por: $f_H(t) = \max_{0 \le t' \le t} f_C(t')$, onde

$$\mathbf{w}_H = \mathbf{w}_H, \text{ para } f(\mathbf{w}_C(t')) = f_H(t) \text{ e } 0 \le t' \le t.$$

- Antítese absoluta: O vetor antítese absoluta do polo w é definido pelo seu vetor oposto expresso como $\breve{\mathbf{w}}$. A operação de antítese absoluta define-se como: Dado um x tal que $a \le x \le b$, onde $a, b \in \mathbb{R}$, o oposto de x é dado por (RAHNAMAYAN; TIZHOOSH; SALAMA, 2007): $\breve{x} = b - x + a$. Assumindo $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ e que $\mathbf{x} \in S \Rightarrow r_i \le x_i \le s_i$, $\forall$i = 1, 2, …, n, onde r_i e s_i são os limites inferior e superior da i-ésima dimensão de S.
- Contradição: Sendo $d: S^2 \rightarrow \mathbb{R}_+$ uma função distância de forma que $\delta_{p,q} = \delta_{q,p}, \forall\, p, q$. Um exemplo típico é a função de distância euclidiana.
- Síntese: Segundo à definição da dialética, a síntese representa a resolução da contradição existente entre dois polos, onde um exerce o papel de tese e outro de antítese (BORNHEIM, 1977). De forma intuitiva, a resolução de uma contradição a partir de uma função de síntese deveria considerar que todas as sínteses encontradas herdassem características de todos os polos que compõe a contradição (BORNHEIM, 1977; GRAMSCI, 1989; KONDER, 2004; VAZQUEZ, 2007). Um exemplo de síntese que segue tal abordagem, a qual foi utilizada

neste trabalho é definido a seguir. Seja *g* uma função da forma $g:S^2 \rightarrow S^2$ que se aplicada a dois polos quaisquer pertencentes ao espaço de busca de soluções, $\mathbf{w}_p$ e $\mathbf{w}_q$, produz dois outros polos, $\mathbf{w}_u$ e $\mathbf{w}_v$, cujas componentes são definidas como a seguir:

$$w_{u,i} = \begin{cases} w_{p,i}, & i \bmod 2 = 0, \\ w_{q,i}, & i \bmod 2 = 1, \end{cases},$$
$$w_{v,i} = \begin{cases} w_{q,i}, & i \bmod 2 = 0, \\ w_{p,i}, & i \bmod 2 = 1, \end{cases},$$

para i = 1, 2, …, *n*. Um exemplo gráfico do caso onde a resolução de uma contradição entre uma tese e uma antítese, ambas bidimensionais, que segue esta definição é apresentado na Figura 5.1.

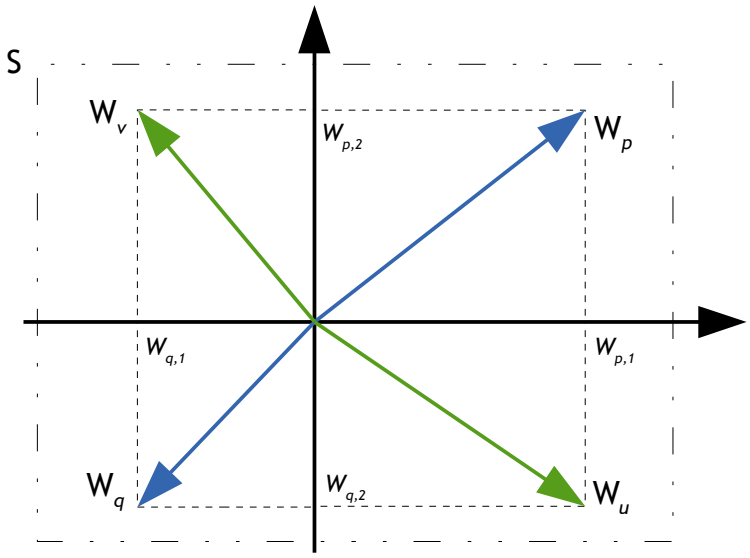

Figura 5.1: Síntese aplicada aos polos wp e wq produzindo dois polos sínteses, wu e wv. Ambas sínteses wu e wv possuem características de ambos polos, wp e wq

Um outro exemplo de síntese que seguiria a mesma ideia seria uma função da forma $g:S^2 \rightarrow S$, onde cada componente do único polo síntese produzido, $\mathbf{w}_u$, é definida como a média aritmética entre as respectivas componentes dos polos tese e antítese, $\mathbf{w}_p$ e $\mathbf{w}_q$, conforme a seguir:

$$w_{u,i} = (w_{p,i} + w_{q,i})/2.$$

5.3 ALGORITMO DE BUSCA E OTIMIZAÇÃO

O algoritmo que adapta o método dialético para busca e otimização é descrito em (DOS SANTOS, 2009) como:

1. Inicialmente definem-se a quantidade inicial $m(0)$ de polos integrantes do sistema dialético $\Omega(0)$, o número de fases históricas, n_P, e a duração de cada fase histórica, n_H. Esses são parâmetros do método dialético de otimização. O número inicial de polos corresponde ao conceito de população inicial das abordagens evolucionárias tradicionais e deve ser par, de forma que a metade do número de polos seja gerada de forma aleatória, dentro do domínio da função objetivo, enquanto a outra metade é obtida pela geração de polos antítese absoluta. Este processo gera uma luta de polos mais intensa, gerando uma maior dinâmica inicial.
2. Enquanto não se atinge um máximo de n_P fases históricas e a força hegemônica histórica não é maior do que um dado limiar superior de força (estimativa inicial do valor máximo da função objetivo), $f_H(t) < f_{sup}$ (critério para se considerar o máximo da função objetivo atingido), repete-se:
   i. Evolução: Enquanto não se atinge um máximo de n_H iterações e $f_H(t) < f_{sup}$, os polos são ajustados segundo a seguinte expressão:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \Delta\mathbf{w}_{C,i}(t) + \Delta\mathbf{w}_{H,i}$$

para

$$\begin{gathered}\Delta\mathbf{w}_{C,i} = \eta_C(t)(1-\mu_{C,i}(t))^2(\mathbf{w}_C(t)-\mathbf{w}_i(t)),\\ \Delta\mathbf{w}_{H,i} = \eta_H(t)(1-\mu_{H,i}(t))^2(\mathbf{w}_H(t)-\mathbf{w}_i(t)),\\ 0<\eta_C(t)<1,\\ 0<\eta_H(t)<1,\end{gathered}$$

onde $\eta_C(0) = \eta_H(0) = \eta_0$ e $0 < \eta_0 < 1$. Os termos $\Delta\mathbf{w}_{C,i}(t)$ e $\Delta\mathbf{w}_{H,i}(t)$ modelam as influências das hegemonias, presente e histórica respectivamente, sobre o i-ésimo polo. Os termos $\eta_C(t)$ e $\eta_H(t)$ correspondem aos passos de atualização dos polos, presente e histórica, que são

atualizados em cada iteração e a cada fase histórica, respectivamente, conforme

$$\eta_C(t+1)=\alpha\,\eta_C(t)$$

ao final de cada iteração e

$$\eta_H(t+1)=\alpha\,\eta_H(t)$$

ao final de cada fase histórica, para $\alpha < 1$ (tipicamente, $\alpha = 0{,}9999$). Embora lento, o decremento dos passos ao longo das iterações tem por objetivo facilitar a convergência do algoritmo, assim como ocorrem em redes neurais (EBERHART; SHI, 2007). Os termos $\mu_{C,i}(t)$ e $\mu_{H,i}(t)$ representam, respectivamente, o grau de pertinência presente e o grau de pertinência histórica e baseados nas funções de pertinência da versão clássica do classificador não-supervisionado c-médias nebuloso (ZHU; JIANG, 2003) conforme a seguir:

$$\mu_{C,i}(t)=\left(\sum_{j=1}^{m}\frac{|f(\mathbf{w}_i(t))-f_C(t)|}{|f(\mathbf{w}_j(t))-f_C(t)|}\right)^{-1},$$
$$\mu_{H,i}(t)=\left(\sum_{j=1}^{m}\frac{|f(\mathbf{w}_i(t))-f_H(t)|}{|f(\mathbf{w}_j(t))-f_H(t)|}\right)^{-1},$$

onde $1 < i < m(t)$. Assim, quando a força social do i-ésimo polo, $f(\mathbf{w}_i(t))$, for próxima da força hegemônica presente, $f_C(t)$, o termo $\mu_{C,i}(t)$ será próximo de 1, fazendo a influência da hegemonia presente, $\Delta\mathbf{w}_{C,i}(t)$, próxima de zero, praticamente anulando seu efeito. Quando $f(\mathbf{w}_i(t))$, se aproximar do valor da força hegemônica histórica, $f_H(t)$, o comportamento é similar.

ii. Crise Revolucionária: Neste estágio, os seguintes passos são executados:

1. Todas as contradições, $\delta_{i,j}$, são avaliadas e, a partir delas, é definido um limiar chamado contradição mínima, [10]δ_{min}. Todos os polos envolvidos com contradição abaixo deste limiar serão fundidos em um único polo. Esse processo produzirá um novo conjunto de polos,

[10] Neste trabalho, a contradição mínima,δ_{min}, foi definida como a contradição média subtraída de um desvio-padrão.

$\Omega(t+1)$ formado pela união do subconjunto dos polos fusão com subconjunto dos demais polos envolvidos em contradições acima do limiar mínimo conforme abaixo:

$$\delta_{i,j}(t) > \delta_{min} \Rightarrow \mathbf{w}_i(t), \mathbf{w}_j(t) \in \Omega(t+1),$$
$$\delta_{i,j}(t) \leq \delta_{min} \Rightarrow \mathbf{w}_i(t) \in \Omega(t+1),$$

com $i \neq j, \forall i, j$ onde $1 \leq i \leq m(t)$ e $1 \leq j \leq m(t)$.

2. Das contradições calculadas na etapa anterior, aquelas de maior valor serão consideradas como as *principais contradições do sistema dialético*[11]. Os polos envolvidos em tais contradições são considerados pares tese-antítese. Tais pares de polos unidos aos seus respectivos polos sínteses, terminam de completar o novo conjunto de polos, $\Omega(t+1)$.
3. O efeito de crise é adicionado a todos os polos no sistema dialético, $\Omega(t+1)$, gerando um novo conjunto de polos, $\Omega(t+2)$, para $\mathbf{w}_k(t+1) \in \Omega(t+1)$ desde que:

$$w_{k,i}(t+2) = w_{k,i}(t+1) + X_{max} \cdot G(0,1)$$

para $1 \leq k \leq m(t)$ e $1 \leq i \leq n$, onde G(0,1) é um número aleatório extraído de uma distribuição de Gauss com valor esperado igual a zero e variância igual a 1.

4. Se o critério de parada não é atingido, um novo conjunto de polos é gerado conforme a seguir:

$$\mathbf{w}_i(t+2) \in \Omega(t+2) \Rightarrow \breve{\mathbf{w}}_i(t+2) \in \Omega(t+2)$$

para $1 \leq i \leq m(t+2)$, onde $m(t+2) = 2\mathrm{m}(t+1)$. Dessa forma, o conjunto de polos é aumentado em duas vezes através da adição de polos em antítese antagônica aos outros polos existentes anteriormente, repetindo assim a estratégia usada na criação da população inicial. O fluxograma básico do ODM é encontrado na Figura 5.2. Apesar de possuir uma heurística mais complexa que o PSO e o EP, o ODM pode apresentar um custo computacional inferior, uma vez que a solução se aproxime do ótimo ou sub-ótimo global, menores serão as contradições e tal fato pode impactar na taxa de fusão de polos e, consequentemente, no tamanho do conjunto de polos manipulado.

[11] Neste trabalho, o limiar de contradição máxima, δ_{max}, é definido pela média das contradições somada a um desvio-padrão.

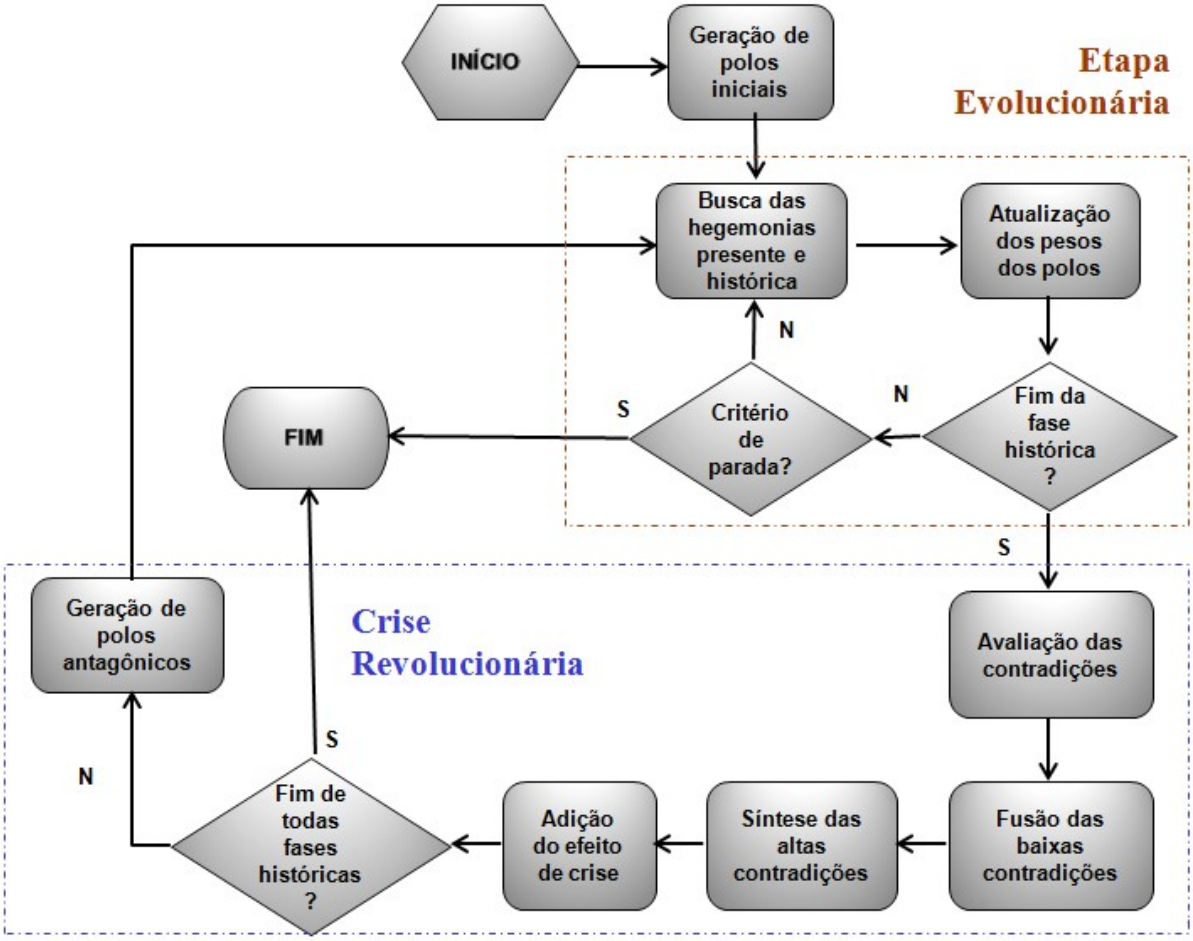

Figura 5.2: Fluxograma básico do ODM

6 ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS USANDO OTIMIZAÇÃO DIALÉTICA E ALINHAMENTO MÚLTIPLO DE SEQUÊNCIAS USANDO ALGORITMOS GENÉTICOS

O alinhamento múltiplo de sequências é considerado um dos mais importantes problemas em aberto da bioinformática. Isso se dá principalmente pela (1) sua capacidade de obter informações biológicas relevantes, indispensáveis em diversas aplicações nas áreas de biologia molecular (NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003; THOMPSON, JULIE D *et al.*, 2011), e pelo (2) aumento da necessidade de processamento eficiente devido à enorme e crescente volume de dados de sequências disponibilizados pelos sequenciadores de última geração (DAUGELAITE; O' DRISCOLL; SLEATOR, 2013).

A extração de informações de relevância biológica exige a utilização de algoritmos eficientes e robustos capazes de manipular esta enorme quantidade de dados. Segundo à literatura, algoritmos baseados no método progressivo (HIGGINS; SHARP, 1988b; LARKIN *et al.*, 2007; SIEVERS *et al.*, 2011) constituem a classe de algoritmos de alinhamento cujo desempenho é mais adequado para lidar com grandes volumes de dados. Seu bom desempenho é proveniente, em especial, de sua heurística de busca na qual é utilizado um método direto (conhecido por programação dinâmica) para realização de diversos alinhamentos de pares sequências que são posteriormente utilizados na construção de um alinhamento múltiplo final. Além disso, a heurística progressiva pode ser acelerada através de técnicas de paralelismo por possuir diversas sub-rotinas otimizáveis por técnicas de paralelismo (LLOYD; SNELL, 2011).

Entretanto, o método progressivo original, implementado pela primeira vez no algoritmo Clustal, possui em sua própria definição um problema de precisão por não permitir reajustes em alinhamentos parciais realizados anteriormente (NOTREDAME; HIGGINS; HERINGA, 2000; NOTREDAME, 2002; SIMOSSIS; KLEINJUNG; HERINGA, 2003). Tal questão é conhecida pela regra *"once a gap always a gap*" que no contexto quer dizer "uma vez inserida, uma lacuna nunca é

removida ou deslocada". Além disso, algoritmos baseados no método progressivo original também apresentam normalmente uma baixa precisão quando aplicados a sequências de baixa similaridade (NOTREDAME, 2002; SIEVERS *et al.*, 2011; SIMOSSIS; KLEINJUNG; HERINGA, 2003). Como não há garantias da existência de um único ótimo global em um alinhamento de sequências baseado em programação dinâmica (NEEDLEMAN; WUNSCH, 1970), a natureza determinística dos métodos progressivos pode limitar o usuário da ferramenta de MSA de conhecer apenas uma única dentre inúmeras opções de alinhamentos ótimos possíveis para um dado conjunto de sequências.

A literatura apresenta algoritmos baseados em adaptações ao método progressivo original, capazes de mitigar ou resolver a maioria dos problemas associados a tal método. Um desses algoritmos, o T-Coffee (NOTREDAME; HIGGINS; HERINGA, 2000) através de alinhamentos múltiplos realizados por consistência promoveu melhorias significativas na precisão (na ordem de 5% a 10% quando comparado ao Clustal) para sequências de baixa similaridade. Além disso, o T-Coffee também foi capaz de eliminar o problema do reajuste em alinhamentos parciais já realizados porém, tudo isso ao preço de um aumento do custo computacional. Além do T-Coffee, outros algoritmos obtiveram resultados semelhantes em tratar os problemas do método progressivo original dos quais destaca-se o MAFFT (KATOH; STANDLEY, 2013), o Kalign (LASSMANN; SONNHAMMER, 2005), o MUSCLE (EDGAR, 2004) e o Clustal Omega (SIEVERS *et al.*, 2011). Entretanto, devido à mesma característica determinística presente em todos, tais algoritmos ainda restringem o usuário da ferramenta de MSA a conhecer apenas um único alinhamento ótimo ou sub-ótimo.

Métodos iterativos são baseados na ideia de que a solução para um dado problema pode ser computada a partir de modificações realizadas em soluções sub-ótimas já conhecidas onde cada ocorrência de modificação é uma iteração (NOTREDAME, 2002). Os métodos iterativos de MSA podem ser classificados segundo suas naturezas estocásticas. Assim, existem métodos iterativos

estocásticos e métodos iterativos não-estocásticos. No segundo caso, tratam-se de métodos, normalmente progressivos, que definem alguma sub-rotina interativa de poucos ciclos (normalmente de apenas duas iterações) para refinamento da solução. Já os métodos iterativos estocásticos diferem-se completamente dos métodos progressivos por apresentarem uma natureza estocástica de busca a qual permite a um algoritmo baseado neste método, encontrar (caso existam) soluções ótimas ou sub-ótimas diferentes entre si – bastando que tal algoritmo seja executado mais de uma vez. Além disso, algoritmos iterativos de alinhamento podem ser capazes de promover melhorias (DE SOUZA; DOS SANTOS; YARA, 2013) em soluções encontradas por métodos progressivos (LARKIN *et al.*, 2007) de forma significativa. Este trabalho segue essa ideia ao aplicar um método iterativo de otimização, como algoritmos genéticos ou otimização dialética, para realizar o alinhamento múltiplo de sequências fazendo uso de soluções encontradas pelo ClustalW, como também para buscar por alinhamentos ótimos de forma independente.

Este capítulo está organizado conforme a seguir: as seções 6.1, 6.2, e 6.3 tratam, respectivamente, das definições a respeito da base de dados usada, da modelagem sugerida e materiais e métodos utilizados. A seção 6.4 apresenta os resultados dos experimentos e na seção Erro: Origem da referência não encontrada são realizados comentários sobre os resultados obtidos, as dificuldades encontradas e sugestões de trabalhos futuros.

6.1 BASE DE DADOS UTILIZADA

Neste trabalho, com desejo de inicialmente validar o modelo proposto de forma computacional, optou-se pela construção sintética de uma base de dados utilizando como base sequenciamentos reais. A ideia foi construir uma base de 50 grupos de quatro sequências onde cada uma dessas é obtida aleatoriamente de um genoma diferente.

As sequencias usadas correspondem às primeiras 960 bases dos genomas AAXI01000001 e AAXI01000040 da base pública do EMBL-EB e correspondem a sequenciamentos do genoma do fungo *Pyrenophora tritici-repentis* Pt-1C-BFP (BIRREN *et al.*, 2007) o agente patogênico da

mancha amarela, chamada de *tan spot*, uma das doenças de maior impacto econômico que afetam o trigo em lavouras no Brasil como também em todo o mundo (FRIESEN; FARIS, 2004; SANTANA; CLEBSCH; FRIESEN, 2008).

Para reduzir a carga computacional durante os experimentos, optou-se em utilizar pequenos segmentos de 60 bases no lugar das longas sequências de 960 bases. Nesta estratégia, cada sequência de 960 bases foi dividida em 16 segmentos de comprimento 60. Assim, durante a construção de cada grupo, foi escolhida uma posição, dentre as 16 possíveis, para extrair os segmentos de 60 bases que formarão cada uma das 4 sequências do grupo.

Após os 50 grupos de 4 sequências serem definidos, cada um foi alinhado através da ferramenta Clustal W2 disponível no website do EMBL-EBI utilizando os parâmetros padrões da ferramenta, conforme a Tabela 6.1. Estes alinhamentos foram salvos para posteriormente para: (1) serem avaliados pela mesma função objetivo a foi aplicada a todos os alinhamentos encontrados pelo ODM, (2) semear a população inicial em alguns experimentos e (3) fornecer sugestão de qual número de lacunas utilizar para o alinhamento de cada grupo de sequências.

Tabela 6.1: Valores dos parâmetros utilizados no ClustalW

Nome do Parâmetro	Valor Utilizado
dnamatrix	IUB
gapopen	-10
noendgaps	no
gapext	-0,2
gapdist	5
iteration	none
numiter	1

6.2 MODELAGEM

De forma semelhante à demais classes de algoritmos evolucionários, para responder de forma desejada o ODM requer uma definição correta para (1) o mapeamento entre o espaço de busca do problema e o espaço de busca do algoritmo, através da modelagem da representação (ou

genótipo), como também para (2) a avaliação das soluções candidatas à ótima, representadas por polos, através da definição da função objetivo. Além disso, por se tratar de um método de otimização para casos contínuos, ODM sofreu algumas adaptações para ser utilizado no universo discreto do problema do MSA modelado por posicionamento das lacunas. Uma destas modificações foi a definição de um limiar (0,5) para atualização ou não dos pesos por influência das hegemonias (presente e histórica). A crise revolucionária foi modelada como uma pertubação que poderia ocorrer de duas formas: *random resetting* ou *creep mutation*. Estudos empíricos realizados durante trabalho demonstraram que operador *random resetting* produzia resultados melhores que o *creep mutation*. Mais detalhes sobre os parâmetros do método dialético de otimização aplicado são apresentadas na seção 6.3.1

6.2.1 Candidatos à solução

Neste trabalho, uma nova modelagem dos candidatos à solução para o problema de MSA é sugerida como uma definição orientada à posição das lacunas (*gaps*) nas sequências. Nesta modelagem, polos são formados por vetores de números inteiros que correspondem às posições em cada sequência onde estão posicionadas as lacunas e representam os candidatos à solução. Não há necessidade de incluir na estrutura do candidato à solução o conjunto de sequências a serem alinhadas pois este conjunto é imutável para qualquer alinhamento construído.

Assim, a definição formal de um alinhamento múltiplo de sequências neste trabalho corresponde a uma tupla formada por (1) uma matriz $\mathbf{M}_{N,\lambda}$ na qual cada uma das N linhas corresponde a uma sequência de comprimento λ, e por (2) um polo **w**, o qual corresponde as posições das lacunas em cada uma das N sequências do alinhamento. Assim, o i-ésimo candidato a solução corresponde à

$$\mathbf{w_i} = (w_{i,0,0}, w_{i,0,1}, \ldots, w_{i,0,L}, w_{i,1,0}, w_{i,1,1}, \ldots, w_{i,1,L}, \ldots, w_{i,N,0}, w_{i,N,1}, \ldots, w_{i,N,L},)^T,$$

com $1 \leq w_{i,n,m} \leq \lambda + L$ onde $w_{i,n,m}$ é corresponde à posição da m-ésima lacuna inserida na n-ésima

sequência com $m \in \{1, 2, \ldots, L\}$, $n \in \{1, 2, \ldots, N\}$, de modo que L corresponde ao total de lacunas inseridas em cada sequência; λ é o comprimento original de cada sequência sem a adição de lacunas, $T = N.L$ corresponde ao total de lacunas usadas no alinhamento e também à dimensionalidade de $\mathbf{w}_i$. Exemplso de alinhamentos pela representação sugerida podem ser vistos na sexta coluna (Polos Exemplo) da Tabela 6.2.

6.2.2 Definição da quantidade de lacunas utilizada

Na modelagem adotada, a escolha do número de lacunas utilizadas é determinante para a definição da estrutura dos polos. Para um dado conjunto de sequências, o uso de um número de lacunas além do suficiente para produzir um alinhamento ótimo provocaria um aumento no comprimento do vetor de posições e isso implicaria em um maior custo computacional. No caso de optar-se por uma quantidade de lacunas inferior à necessária, a capacidade de busca pode ser comprometida. Assim, a dificuldade na definição da quantidade de lacunas a serem utilizadas se demonstra um problema não trivial cuja análise foi realizada experimentalmente sob três abordagens. Na primeira abordagem, chamada Canônica, foi utilizada uma quantidade de lacunas que correspondia a 50% do comprimento original das sequências o qual foi de 60 bases. Em uma segunda abordagem, chamada NLC (Número de Lacunas Clustal) para cada conjunto de sequências foi utilizada a quantidade correspondente de lacunas utilizada pelo ClustalW. Por fim, na terceira abordagem, chamada de NLF (Número de Lacunas Flutuante) a quantidade de lacunas disponíveis para cada sequência foi mantida (50% do comprimento), porém a inicialização dos valores (durante a inicialização da população) foi modificada seguindo um esquema onde, para cada sequência, apenas uma parcela das lacunas recebe valores de posições escolhidos ao acaso, enquanto que o restante das posições recebe um valor que já fora usado na mesma sequência. Assim, a quantidade de lacunas realmente inseridas em cada alinhamento candidato variava entre 5% e 50% do comprimento (λ) das sequências. Os limiares máximo (50%) e mínimo (5%) foram baseados na variação da quantidade de lacunas utilizadas pelo Clustal ao longo dos experimentos. Apesar desta

modificação durante a inicialização da população, o processo de atualização dos pesos se manteve e, portanto, os alinhamentos candidatos poderiam ter seus valores iniciais reajustados e, consequentemente, a quantidade de lacunas inseridas poderia ser guiada segundo à evolução. A Tabela 6.2 apresenta exemplos de polos utilizados em cada uma das três abordagens.

Tabela 6.2: Exemplos de polos gerados na população inicial para cada uma das três abordagens onde λ corresponde ao comprimento das *N* sequências a serem alinhadas, *L* o total de posições de lacunas a serem inicializadas e *L Clustal* a quantidade de lacunas utilizadas pelo Clustal

Abordagem	λ	*L*	*N*	*L Clustal*	Polos exemplo
Canônica	6	50% de λ	3	2	(1,4,3, 2,3,4, 0,1,2) (5,4,0, 0,3,1, 3,6,0)
NLC	6	L Clustal	3	2	(2,3, 4,1, 0,1) (1,4, 0,1, 1,2)
NLF	6	entre 5% e 50% de λ	3	2	(1,4,4, 2,1,1, 5,3,3) (0,0,0, 3,3,3, 1,1,1) (3,4,1, 5,0,1, 1,2,3)

6.2.3 Função objetivo

Neste trabalho, a função objetivo, responsável por avaliar matematicamente a qualidade biológica de um alinhamento múltiplo, é definida pela composição de outras quatro funções mais simples, onde cada uma busca atender um aspecto específico do problema de MSA. A ideia dessa modelagem é premiar aspectos desejáveis e penalizar os não-desejáveis.

6.2.3.1 Pontuação por Similaridades

O primeiro aspecto considerado pela função objetivo trata da necessidade de atribuir pesos às correspondências identificadas em uma mesma coluna. Este esquema de pontuação, conhecido por *matching score,* baseia-se no modelo da função *sum-of-pairs* (NEEDLEMAN; WUNSCH, 1970) no qual é calculado o somatório, para cada coluna, de todas as correspondências entre bases de uma mesma coluna. O peso de cada correspondência é determinado através de uma matriz de similaridades $M(i \times j)$ na qual cada elemento $m_{i,j}$ representa a similaridade entre a *i*-ésima e a *j*-ésima base. Neste trabalho, a matriz de similaridade utilizada foi a matriz IUB, que corresponde à matriz

padrão usada pelo ClustalW em alinhamento de sequências de DNA. Na matriz IUB, todos os elementos da diagonal principal, os quais correspondem às correspondências ou *matches*, são preenchidos com 1,9 enquanto que os elementos restantes, conhecidos como não-correspondências ou *mismatches* são preenchidos com zeros.

6.2.3.2 Penalidade pelo uso de lacunas

O segundo aspecto considerado na definição da função objetivo leva em conta o posicionamento das lacunas ao longo das sequências. De forma semelhante a outros métodos de alinhamento, a função objetivo deste trabalho trata a necessidade de desencorajar o mau posicionamento de lacunas através da aplicação de uma função de penalidade.

A inserção de lacunas está relacionada à representação da ocorrência de dois tipos mutação: por inserção e por exclusão. Neste primeiro tipo, a mutação é representado geralmente pelo posicionamento de diversas lacunas em uma mesma coluna enquanto que para representar uma mutação do tipo exclusão geralmente é utilizada uma única lacuna. É interessante que a função objetivo leve em conta tais observações para que um viés contra mutações de inserções não seja inserido em sua definição.

Neste trabalho, a componente da função objetivo responsável por penalizar o mau uso de lacunas é dividida em duas sub-funções: uma busca penalizar o uso de lacunas de acordo com a sua posição ao longo de uma sequência (posicionamento horizontal) enquanto que outra promove penalidades de acordo com a quantidade de lacunas usadas na mesma coluna (posicionamento vertical).

A penalidade pelo posicionamento horizontal de uma lacuna leva em conta a posição à esquerda. Assim, caso à esquerda da lacuna exista uma base então tal lacuna dá início a uma abertura de espaço (chamado de *gap open*). Caso contrário, ou seja, se na posição à esquerda existe outra lacuna então trata-se de uma extensão desta. Neste trabalho, lacunas do tipo extensão são penalizadas em -0,2 enquanto que lacunas do tipo abertura são penalizadas com -10.

A penalidade por posicionamento vertical leva em conta apenas a quantidade de lacunas

posicionadas em outras sequências na mesma posição. Este trabalho propôs um mecanismo no qual a penalidade diminui exponencialmente com o aumento da quantidade de lacunas na mesma posição.

6.2.3.3 Pontuação pelo número de correspondências na mesma coluna

Em métodos de otimização iterativos, como o método dialético, o processo de busca pela solução ótima é guiado pela evolução gradual dos candidatos ocorrida ao término de cada iteração (NOTREDAME, 2002). Com o desejo de melhor guiar o processo evolucionário do método de otimização aplicado, a definição da função objetivo considera um terceiro aspecto o qual trata da necessidade de premiar alinhamentos à medida que cresce, em cada coluna, o total de correspondências encontradas. Para isso, foi criada uma outra função de pontuação que atribui uma pontuação extra a qual cresce exponencialmente em relação ao número de correspondências ocorridas em uma mesma coluna.

6.2.3.4 Pontuação pela consecutividade de colunas com alta quantidade de correspondências

O quarto aspecto levado em conta pela função objetivo trata da necessidade de privilegiar alinhamentos nos quais colunas de alta similaridade ocorram consecutivamente. Tal necessidade é motivada pelo desejo de identificar regiões de alta similaridade pois estas correspondem a indícios mais fortes de existência de homologia entre as sequências (NOTREDAME; HOLM; HIGGINS, 1998).

Assim, a função objetivo proposta aplicada ao *i*-ésimo candidato a solução f definida como:

$$f_{obj}(\mathbf{w}_i) = \omega_{SP} \cdot f_{SP}(\mathbf{w}_i) + \omega_{Penal} \cdot f_{Penal}(\mathbf{w}_i) + \omega_{Corresp} \cdot f_{Corresp}(\mathbf{w}_i) + \omega_{Consec} \cdot f_{Consec}(\mathbf{w}_i),$$

onde ω_{SP}, ω_{Penal}, $\omega_{Corresp}$ e ω_{Consec} são números reais. As funções componentes são definidas a seguir:

i. $f_{SP}(\mathbf{w}_i)$ corresponde à função para cálculo das similaridades e é definida como

$$f_{SP}(\mathbf{w}_i) = \sum_{s=1}^{S} \sum_{n=2}^{N} IUB(W_{1,s}, W_{n,s})$$

com $W_{n,s}$ correspondendo ao elemento localizado na *n*-ésima linha da *s*-ésima coluna da matriz *W* a qual representa o alinhamento produzido pela inserção das lacunas nas posições indicadas em $\mathbf{w}_i$;

ii. $f_{Penal}(\mathbf{w}_i)$ é a função para o cálculo das penalidades pela inserção das lacunas. O valor da penalidade atribuída para cada lacuna é soma entre a penalidade relativa à sua posição dentro da sequência, chamada de penalidade sob posicionamento horizontal, e a penalidade relativa à quantidade de lacunas existentes em outras sequências que sofreram penalização horizontal na mesma posição, chamada de penalidade sob posicionamento vertical, sendo definida como

$$f_{Penal}(\mathbf{w}_i) = w_V * penalVert(\mathbf{w}_i, S) + w_H * penalHoriz(\mathbf{w}_i, S),$$

onde w_H corresponde ao peso da componente de penalidade horizontal, $penalHoriz(\mathbf{w}_i, S)$, definida por

$$penalHoriz(\mathbf{w}_i, S) = \sum_{n=1}^{N} \sum_{s=1}^{S} gap_{penalH}(\mathbf{w}_i, n, s),$$

com

$$gap_{penalH}(\mathbf{w}_i, n, s) = \begin{cases} 0, & (s=1) \vee (s < p_{FB}(\mathbf{w}_i, n)) \vee (s > p_{LB}(\mathbf{w}_i, n)) \\ penal_{gapExt}, & (s>1) \wedge gap_{penalH}(\mathbf{w}_i, n, (s-1)) < 0 \\ penal_{gapOpen}, & (s>1) \wedge gap_{penalH}(\mathbf{w}_i, n, (s-1)) \geq 0 \end{cases},$$

que w_V corresponde ao peso da componente de penalidade vertical, $penalVert(\mathbf{w}_i, S)$, definida por

$$gap_{penalV}(\mathbf{w}_i, S) = \sum_{s=1}^{S} \frac{penal_{gapOpen}}{columnGaps(\mathbf{w}_i, s)},$$

onde $columnGaps(\mathbf{w}_i, s)$ corresponde à quantidade de lacunas posicionadas na *s*-ésima posição do *i*-ésimo alinhamento e que sofreram penalização horizontal com $0 \leq columnGaps(\mathbf{w}_i, s) \leq N$; $p_{FB}(\mathbf{w}_i, n)$ e $p_{LB}(\mathbf{w}_i, n)$ correspondem respectivamente às posições da primeira e última bases posicionadas na *n*-ésima sequência do *i*-ésimo alinhamento, as constantes $penal_{gapOpen}$ e $penal_{gapExt}$ correspondem respectivamente às

penalidades por abertura de lacuna e por extensão de lacuna com $penal_{gapOpen}$ = -10 e $penal_{gapExt}$ = -0,2, e as constantes N e S equivalem respectivamente ao número de linhas (ou total de sequências do alinhamento) e o total de colunas da matriz W, a qual representa o alinhamento produzido pela inserção das lacunas nas sequências segundo às posições indicadas em $\mathbf{w}_i$. Os valores das constantes w_V e w_H são determinados empiricamente como 0,2 e 1,0 respectivamente.

iii. $f_{Corresp}(\mathbf{w}_i)$ é a função para pontuação devido à quantidade de correspondências (*matches*) identificadas em cada coluna do i-ésimo alinhamento e é definida como

$$f_{Corresp}(\mathbf{w}_i) = -\frac{N \cdot penal_{gapOpen}}{N - \sum_{s=2}^{S} numCorresp(\mathbf{w}_i, s)},$$

onde $numCorresp(\mathbf{w}_i, s)$ representa o número de correspondências identificadas na s-ésima posição (coluna) do i-ésimo alinhamento com $0 \le numCorresp(\mathbf{w}_i, s) \le (N-1)$;

iv. $f_{Consec}(\mathbf{w}_i)$ é a função que calcula a pontuação relativa à consecutividade de colunas de alta similaridade e é definida por

$$f_{Consec}(\mathbf{w}_i) = \sum_{s=1}^{S} Consec(w_i, s),$$

com

$$Consec(\mathbf{w}_i, s) = \begin{cases} 0, & s = 1 \\ \dfrac{numCorresp(s-1)}{(N - numCorresp(\mathbf{w}_i, s))^2}, & s \ge 1 \end{cases}$$

onde *IUB* é a matriz de similaridades e s e n são respectivamente o número de colunas (posições) e o número de sequências (linhas) do alinhamento, com $1 \le s \le S$ e $2 \le n \le N$ sendo S o total de colunas e N o número de sequências do alinhamento.

v. Os parâmetros ω_{SP}, ω_{Penal}, $\omega_{Corresp}$ e ω_{Consec} são usados para configurar a influência de cada função de pontuação correspondente sobre o valor total da função objetivo. Seus valores, definidos empiricamente, são apresentados na Tabela 6.3.

Tabela 6.3: Valores utilizados para os multiplicadores das funções componentes

Parâmetro	Valor Utilizado
ω_{SP}	1,0
ω_{Penal}	1,0
$\omega_{Corresp}$	2,0
ω_{Consec}	1,0

6.3 EXPERIMENTOS

Os experimentos foram realizados ao longo de 50 ensaios onde, para cada um deles, foi utilizado um grupo de sequências construído sinteticamente conforme discutido na seção 6.1. Em cada ensaio, o conjunto de sequências é alinhado através de três diferentes abordagens e os *scores* obtidos são comparados entre si e com aquele obtido pelo alinhamento sugerido pelo Clustal. As abordagens sugeridas são: (1) Forma Canônica, (2) Número de lacunas do Clustal (NLC) e (3) Número de lacunas flutuante (NLF). Em cada ensaio foram realizados vinte execuções para cada uma das três abordagens. Todos os experimentos foram realizados utilizando dois métodos de otimização: algoritmos genéticos (GA) como também o dialético (ODM). Para cada método, a quantidade máxima de gerações foi 1500.

A cada uma das abordagens, além da versão prevista, também foi realizada a execução de uma variação do algoritmo a qual faz uso da técnica de semeadura da população. Nela, a semente utilizada corresponde ao alinhamento sugerido pelo método ClustalW. A técnica de semeadura além poder promover melhorias nos alinhamentos realizados por ferramentas como o ClustalW, também visa permitir o uso de outros recursos providos por tais ferramentas como a construção da árvore filogenética.

Nos experimentos com o ODM, percebeu-se uma melhor convergência com a diminuição da quantidade de gerações para cada período histórico e com o consequente aumento do número de períodos históricos (e momentos de crise). Para ilustrar este comportamento da convergência com relação ao comprimento (em gerações) de cada período histórico, na seção de Resultados são

apresentados resultados obtidos utilizando-se ODM com 2 gerações por período histórico (totalizando 750 períodos históricos) e também 3 gerações por períodos histórico (totalizando 500 períodos históricos).

6.3.1 Parametrização do ODM

Os valores dos parâmetros específicos ao ODM permaneceram os mesmos para todas as abordagens, enquanto que o parâmetro de número máximo de lacunas variou a cada abordagem conforme discutido na seção 6.2.2. A quantidade de gerações por período histórico teve seu valor ótimo empiricamente definido como 2 gerações para cada um dos 750 períodos históricos. Da mesma forma, estudos empíricos também mostraram que definir os limiares de baixa e alta contradição definidos como um desvio-padrão da contradição média de toda população ao final do período histórico foi uma estratégia adequada para a manutenção da diversidade e do tamanho da população (limiares mal definidos poderiam causar a explosão do tamanho da população devido à operação de síntese de altas contradições ou a redução da população a apenas um único polo devido à operação de fusão de polos de baixa contradição). Para ajudar na manutenção da diversidade, sempre que o tamanho da população caia abaixo de um limiar (20% da população inicial), polos gerados aleatoriamente são inseridos à população até que esta volte a possuir pelo menos de 20% da quantidade inicial de polos.

6.3.2 Parametrização do GA

Os experimentos com algoritmos genéticos utilizaram os valores para os parâmetros deste método de otimização segundo à Tabela 6.4.

Tabela 6.4:Valores utilizados para os parâmetros do GA

Prob. de cruzamento	0,75
Prob. de mutação	0,20
Núm. máximo de gerações	1500
Tamanho da população	120
Tamanho do conjunto elite	10% da população

6.3.3 Abordagem Canônica

Nessa abordagem, são utilizados o algoritmo dialético discreto e o algoritmo genético em suas formas canônicas, ou seja, nenhuma informação prévia sobre os alinhamentos obtidos pelo Clustal é utilizada. Todos os elementos da população inicial têm os valores dos seus pesos aleatoriamente escolhidos de forma que repetições de valores são permitidas. Assim, neste processo, a ocorrência de repetições de posições das lacunas implica, na prática, na mudança da quantidade de lacunas realmente inseridas (para manter a uniformidade do comprimento das sequências, são inseridas lacunas no final de cada sequência até que o tamanho máximo seja atingido). Assim, o total de posições inicializadas é igual ao número de lacunas máximo e corresponde a 50% do tamanho (60 bases) das sequências originais. Na variação dessa abordagem, na qual ocorre a inserção de um indivíduo semente na população, é possível que este indivíduo semente utilize uma quantidade de lacunas inferior ao número máximo de lacunas (30 lacunas) disponível pela modelagem. Nesse caso, as posições não usadas recebem um valor que não produz alterações na estrutura do alinhamento obtido pelo ClustalW (por exemplo, recebem o valor de alguma posição já utilizada na mesma sequência), porém nada impede que tais valores inicialmente repetidos sejam modificados durante a evolução.

6.3.4 Abordagem NLC: Número de Lacunas do Clustal

Nesta abordagem, a informação sobre a quantidade de lacunas utilizada pelo Clustal, no alinhamento de cada ensaio, ou seja, de cada grupo de sequências, é usada para definir o número de

lacunas utilizado nos alinhamentos produzidos pelo método proposto. Na variação desta segunda abordagem, em cada ensaio, também é inserido um indivíduo "semente" na população o qual corresponde à solução encontrada pelo Clustal para o conjunto de sequências em questão. Assim, todos os indivíduos da população (inclusive o indivíduo semente) têm a mesma quantidade de posições de lacunas inicializadas.

6.3.5 Abordagem NFL: Número de Lacunas Flutuante (Variável)

É esperado que o número de lacunas utilizado no alinhamento de um conjunto de sequências acompanhe, de forma inversa, a variação do grau de similaridade existente entre tais sequências pois a presença de lacunas está associada a eventos de exclusão ou inserção de bases. Através da inspeção dos alinhamentos sugeridos pelo Clustal para os conjuntos de sequências construídos neste trabalho, foi identificada uma variação superior a 300% no número de lacunas utilizado, mesmo todas as sequências envolvidas possuindo o mesmo comprimento. Diante disso, também seria esperado que usar uma grande quantidade de lacunas, além da necessária, fatalmente aumentaria a dificuldade de adaptação do algoritmo e, assim, sua convergência.

Portanto, a quantidade de lacunas utilizadas em um alinhamento pode variar bastante e o uso de quantidade próxima da correta é um ponto crítico na busca de boas soluções. A terceira abordagem, NLF, propõe acelerar a busca através da adoção de uma estratégia diferente para a inicialização dos candidatos à solução. Nela, uma quantidade aleatória de posições são inicializadas com valores aleatórios, como na primeira abordagem, entretanto, as posições restantes são inicializadas com valores que não produzam mudanças no alinhamento como valores repetidos (posições já sorteadas) ou o valor da última posição (não produz deslocamentos nos resíduos nem gera penalidades).

6.4 RESULTADOS

A média dos *scores* ao longo dos 50 ensaios obtidos pelo ODM em cada abordagem, como também nas variações destas com uso de sementes do ClustalW, são apresentados na Figura 6.1. Já na Figura 6.2 são apresentados os resultados obtidos pelo GA. Estes últimos foram publicados em artigo aprovado e apresentado no IX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC) em Outubro de 2013, Fortaleza-CE. O artigo é encontrado no APÊNDICE A – Uma abordagem para alinhamento múltiplo de sequências de DNA usando Algoritmos Genéticos e número variável de lacunas.

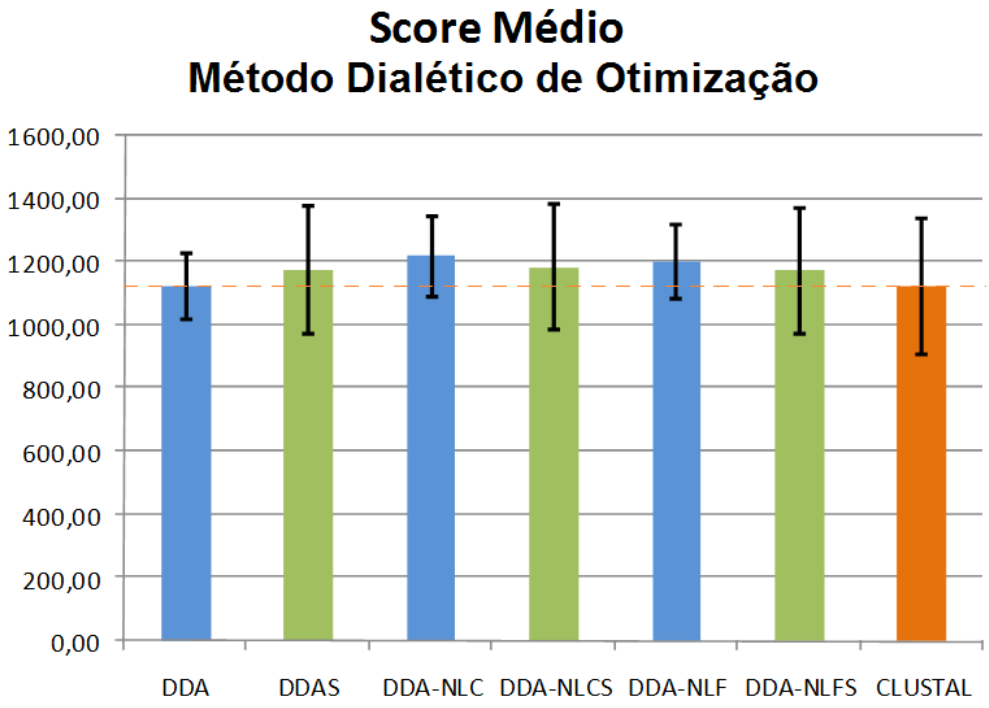

Figura 6.1: Scores obtidos pelo ODM em todas as abordagens e utilizando duas gerações para cada um dos 750 períodos históricos

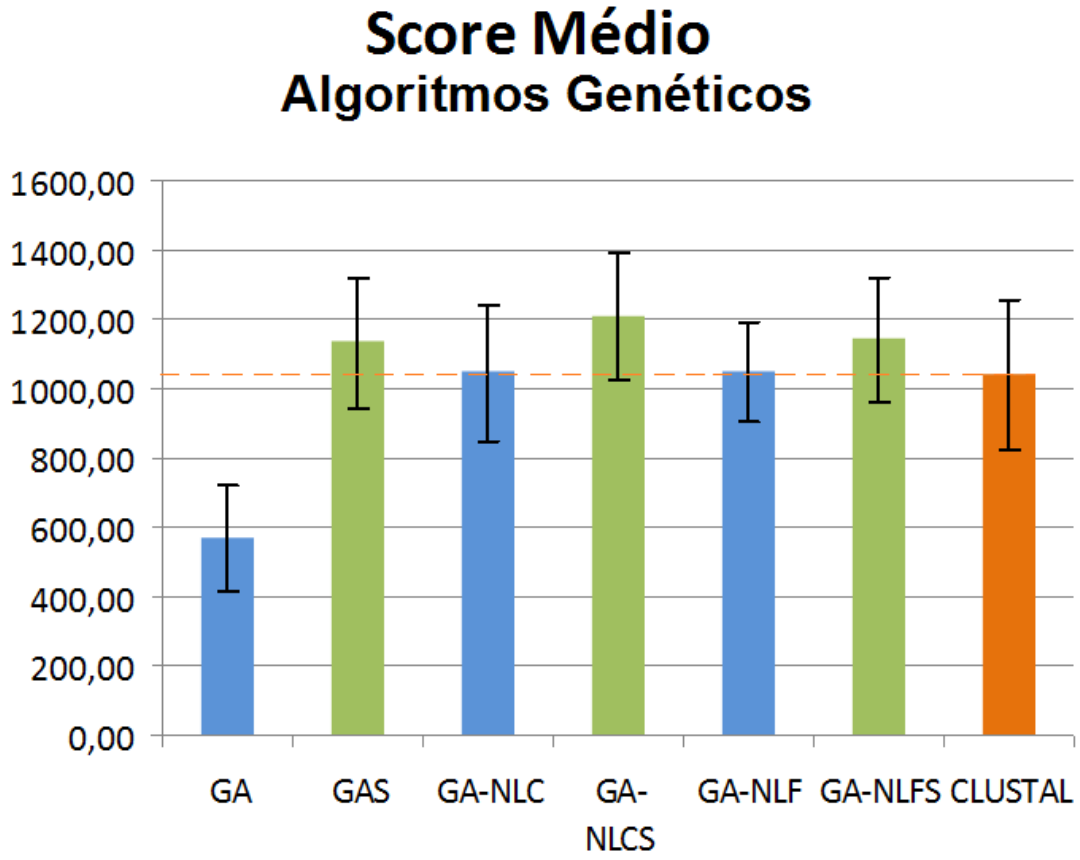

Figura 6.2: Scores obtidos pelo GA em todas as abordagens

6.4.1 Análise estatística

Na Tabela 6.5 (ODM) e na Tabela 6.6 (GA) são apresentados os valores média, máximo, mínimo e desvio padrão para cada uma das três abordagens além do nível de significância obtido através do teste de Wilcoxon para amostras pareadas.

Em um teste de hipóteses, procura-se rejeitar ou não rejeitar a hipótese nula (H_0). Fazendo a hipótese alternativa (H_1) contraditória à nula, têm-se que um valor de *p* abaixo de 0,05 implica na rejeição da hipótese nula (H_0) e, consequentemente, a não rejeição de H_1, enquanto que para *p* maior ou igual a 0,05, a hipótese nula não pode ser rejeitada mas H_1 é rejeitada.

6.4.1.1 Análise dos resultados com algoritmo dialético discreto

A Tabela 6.5 mostra que a hipótese alternativa, H_1, é rejeitada em ambos os casos (H_1:DDA<Clustal e H_1:DDA≠Clustal) na abordagem DDA Canônico, o que demonstra que segundo o teste de Wilcoxon os resultados entre o DDA e o Clustal não apresentam diferenças significativas. Continuando, nas outras duas abordagens (DDA NLC e DDA NLF) é verificado que a hipótese H_1:DDA < Clustal é rejeitada porém H_1:DDA ≠ Clustal não pode ser rejeitada, o que implica que os resultados obtidos pelos DDA são significativamente superiores aos obtidos pelos Clustal.

Tabela 6.5: Valores médios, máximos, mínimos e de significância obtidos pelo ODM em cada abordagem

	Abordagem original							Variação da Abordagem: Uso de Sementes						
		Diferença quando DDA venceu			Diferença quando Clustal venceu				Diferença quando DDA venceu			Diferença quando Clustal venceu		
Abordagem usada	Score DDA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score DDA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal	Score DDA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score DDA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal
DDA Canônico	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1122,59	81,00	27	6,90%	80,37	23	5,73%	1174,01	52,05	16	4,23%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1437,47	482,04	**54%**	41,86%	580,22	**46%**	35,51%	1670,19	617,84	**32%**	47,99%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	833,00	0,00		0,00%	0,00		0,00%	822,31	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	105,45	125,84		10,25%	140,15		9,18%	201,81	120,93		9,26%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,5953510815							**Teste Wilcoxon (valor p): 0,9998014854**						
	H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal							H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal						
	Teste Wilcoxon (valor p): 0,8167875854							**Teste Wilcoxon (valor p): 0,0004824029**						
	H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes							H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes						
DDA Num Lacunas Clustal	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1218,33	149,94	35	11,87%	53,57	15	3,67%	1187,23	117,58	22	8,91%	52,31	16	4,02%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1554,62	589,02	**70%**	46,80%	572,00	**30%**	35,01%	1670,19	678,55	**44%**	41,53%	593,33	**32%**	41,91%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	989,89	0,00		0,00%	0,00		0,00%	814,93	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	126,58	157,35		11,90%	121,39		7,72%	227,62	167,80		11,98%	122,93		9,02%
	Teste Wilcoxon (valor p): 0,99882018							**Teste Wilcoxon (valor p): 0,9788379173**						
	H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal							H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal						
	Teste Wilcoxon (valor p): 0,0024364072							**Teste Wilcoxon (valor p): 0,0438191865**						
	H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes							H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes						
DDA Num Lacunas Flutuante	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1204,02	139,98	33	11,21%	57,93	17	4,00%	1149,61	27,67	10	2,41%	0,02	3	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1467,89	606,13	**66%**	47,52%	504,85	**34%**	30,23%	1670,19	302,39	**20%**	31,11%	0,28	**6%**	0,02%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	925,28	0,00		0,00%	0,00		0,00%	814,93	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	118,07	151,25		11,76%	121,96		7,86%	206,38	73,59		6,43%	0,07		0,01%
	Teste Wilcoxon (valor p): 0,9948822792							**Teste Wilcoxon (valor p): 0,9974152698**						
	H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal							H0: DDA maior ou igual que Clustal :: H1: DDA menor que Clustal						
	Teste Wilcoxon (valor p): 0,0105240118							**Teste Wilcoxon (valor p): 0,0064034604**						
	H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes							H0: DDA e Clustal são iguais :: H1: DDA e Clustal são diferentes						

Os resultados obtidos nas variações com uso de sementes são equivalentes para as três abordagens: a hipótese alternativa sempre é rejeitada no caso H_1:DDA<Clustal, mas não pode ser rejeitada quando H_1:DDA≠Clustal, mostrando que o DDA foi significativamente superior ao Clustal.

6.4.1.2 Análise dos resultados com algoritmos genéticos

Nos resultados obtidos com algoritmos genéticos, apresentados na Tabela 6.6, para a abordagem GA-Canônico, a hipótese alternativa, H_1, não pode ser rejeitada no caso H_1:Clustal<GA, porém é rejeitada para H_1:Clustal≠GA, o que demonstra que, de acordo com o teste de Wilcoxon, os resultados obtidos pelo Clustal são significativamente superiores aos obtidos pelo GA. Nas outras duas abordagens (GA-NLC e GA-NLF) verifica-se que a hipótese alternativa é rejeitada nos casos H_1:Clustal<GA e H_1:Clustal≠GA, o que mostra que os resultados obtidos pelos GA nestas duas últimas abordagens são significativamente equivalentes aqueles obtidos pelos Clustal.

Nas variações com uso de sementes, os resultados obtidos pelo GA são equivalentes para as três abordagens: a hipótese alternativa é rejeitada em ambos os casos H_1:Clustal<GA e H_1:Clustal≠GA, mostrando que os resultados obtidos pelo GA são significativamente superiores aos do Clustal.

Tabela 6.6: Valores médios, máximos, mínimos e de significância obtidos pelo GA em cada abordagem

Abordagem Usada	Abordagem original							Variação da Abordagem: Uso de Sementes						
	Score GA	Diferença quando GA venceu			Diferença quando Clustal venceu			Score GA	Diferença quando GA venceu			Diferença quando Clustal venceu		
		Absoluta	Casos	Percentual % entre Diferença Absoluta e Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal		Absoluta	Casos	Percentual % entre Diferença Absoluta e Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal
GA Canônico	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	567,24	4,80	1	0,63%	480,32	49	44,31%	1132,14	89,38	32	8,00%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	912,51	239,96	**2%**	31,42%	972,22	**98%**	71,64%	1604,63	412,40	**64%**	44,05%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	226,79	0,00		0,00%	0,00		0,00%	754,97	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	149,12	33,94		4,44%	232,59		17,16%	190,94	107,86		9,89%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,9999999994							**Teste Wilcoxon (valor p): 0,0173052788**						
	H0: clustal maior ou igual a GA :: H1: clustal menor que GA							H0: clustal maior ou igual a GA :: H1: clustal menor que GA						
	Teste Wilcoxon (valor p): 5,95073434972449E-010							**Teste Wilcoxon (valor p): 0,9887528644**						
	H0: clustal menor ou igual a GA :: H1: clustal maior que GA							H0: clustal menor ou igual a GA :: H1: clustal maior que GA						
	Teste Wilcoxon (valor p): 1,1901468699449E-009							**Teste Wilcoxon (valor p): 0,0346105575**						
	H0: clustal igual a GA :: H1: clustal diferente de GA							H0: clustal igual a GA :: H1: clustal diferente de GA						
GA Número de Lacunas Clustal (GA-NLC)	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1045,47	114,63	27	9,60%	111,92	23	8,80%	1132,14	89,38	32	8,00%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1442,70	918,98	**54%**	63,70%	1016,58	**46%**	63,35%	1604,63	412,40	**64%**	44,05%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	588,06	0,00		0,00%	0,00		0,00%	754,97	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	195,18	171,49		13,01%	202,23		14,25%	190,94	107,86		9,89%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,6609623499							**Teste Wilcoxon (valor p): 4,17144456013841E-007**						
	H0: clustal maior ou igual a GA :: H1: clustal menor que GA							H0: clustal maior ou igual a GA :: H1: clustal menor que GA						
	Teste Wilcoxon (valor p): 0,3425779004							**Teste Wilcoxon (valor p): 0,999999621**						
	H0: clustal menor ou igual a GA :: H1: clustal maior que GA							H0: clustal menor ou igual a GA :: H1: clustal maior que GA						
	Teste Wilcoxon (valor p): 0,6851558008							**Teste Wilcoxon (valor p): 8,34288912027683E-007**						
	H0: clustal igual a GA :: H1: clustal diferente de GA							H0: clustal igual a GA :: H1: clustal diferente de GA						
GA Número de Lacunas Flutuante (GA-NLF)	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1048,71	94,55	24	8,35%	88,60	26	7,10%	1140,95	98,20	29	8,68%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1336,44	520,79	**48%**	49,86%	539,65	**52%**	37,66%	1604,63	592,96	**58%**	53,10%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	724,41	0,00		0,00%	0,00		0,00%	773,49	0,00		0,00%	0,00		0,00%
	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão	Desvio Padrão		Desvio Padrão	Desvio Padrão		Desvio Padrão
	144,23	138,26		12,11%	133,30		9,79%	177,73	131,61		11,41%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,5499332175							**Teste Wilcoxon (valor p): 1,35118457023899E-006**						
	H0: clustal maior ou igual a GA :: H1: clustal menor que GA							H0: clustal maior ou igual a GA :: H1: clustal menor que GA						
	Teste Wilcoxon (valor p): 0,4538899342							**Teste Wilcoxon (valor p): 0,9999987846**						
	H0: clustal menor ou igual a GA :: H1: clustal maior que GA							H0: clustal menor ou igual a GA :: H1: clustal maior que GA						
	Teste Wilcoxon (valor p): 0,9077798683							**Teste Wilcoxon (valor p): 2,70236914046151E-006**						
	H0: clustal igual a GA :: H1: clustal diferente de GA							H0: clustal igual a GA :: H1: clustal diferente de GA						

6.4.2 Resultados com algoritmo dialético discreto com três gerações por período histórico

Conforme comentado na seção 6.3.1, na parametrização do ODM a melhor distribuição encontrada para as 1500 gerações consistiu em definir períodos históricos limitados a apenas duas gerações. Para demonstrar melhor o efeito do ajuste deste parâmetro (número de gerações por período histórico), todos os experimentos foram refeitos utilizando períodos históricos compostos de três gerações. Os resultados são apresentados na Figura 6.3.

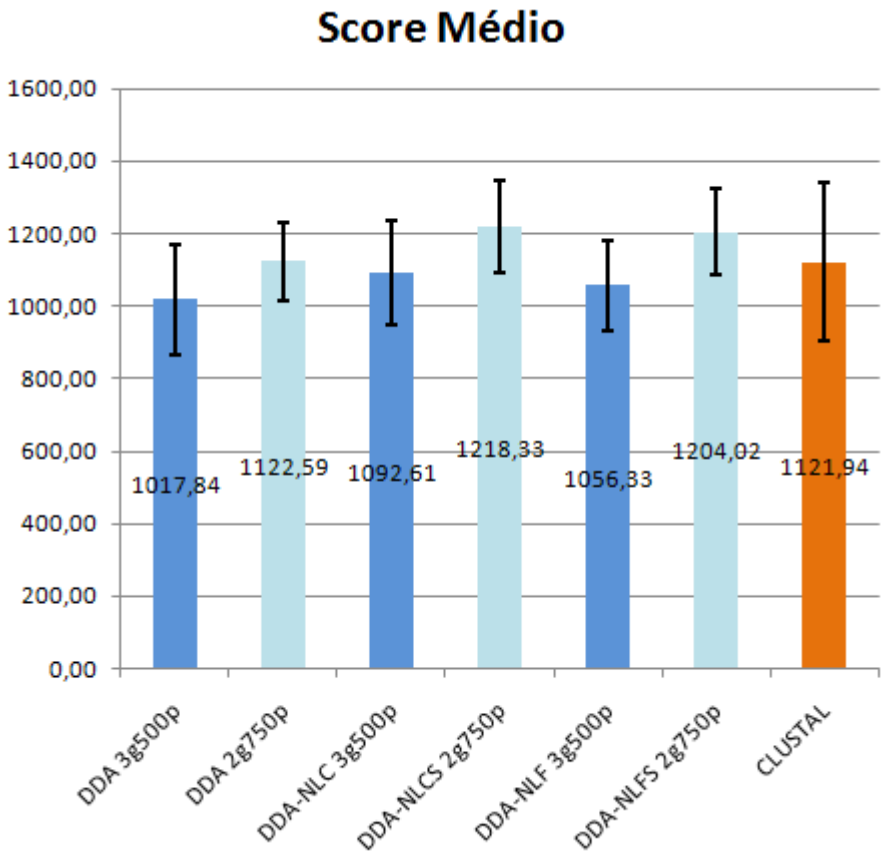

Figura 6.3: Comparativo entre abordagens com diferentes números de gerações por cada período histórico

7 CONCLUSÕES E PERSPECTIVAS

Exceto na abordagem DDA Canônico (sem o uso de sementes), a qual se mostrou equivalente, o método baseado em otimização dialética obteve nas outras duas abordagens resultados significativamente superiores ao ClustalW de acordo com o teste de Wilcoxon para amostras pareadas. Além disso, apesar de apenas significativamente equivalente, a abordagem DDA-Canônico provou ser mais precisa que o ClustalW. O mesmo não ocorreu para a solução baseada em algoritmos genéticos: a análise estatística mostrou que o GA foi inferior ao ClustalW na abordagem GA Canônica e equivalente ao ClustalW de forma significativa nas outras duas abordagens. Já nos resultados com o uso de sementes, ambas técnicas de otimização foram capazes de melhorar significativamente os resultados encontrados pelo ClustalW.

A respeito dos resultados apresentados no comparativo entre a distribuição das 1500 gerações entre os períodos históricos, o fato das abordagens com menos gerações por período histórico (mantendo-se a quantidade total de gerações) terem se demonstrado superiores chama por um debate.

Uma provável causa pode ser explicada pelo formato da superfície de busca. Na modelagem orientada a lacunas, qualquer pequena modificação em uma única lacuna pode deslocar blocos inteiros de resíduos em diversas colunas, com uma probabilidade mais alta de prejudicar bons alinhamento já construídos, pois o *score* seria facilmente reduzido drasticamente. Este comportamento instável sugere uma superfície de busca formada por várias pontas íngremes as quais correspondem a diversos ótimos locais existentes: qualquer mínimo deslocamento provoca uma grande redução do valor da função objetivo. Ainda seguindo esta ideia, o deslocamento dos polos provocado pela influência de polos hegemônicos pode não ser tão eficiente devido a enorme irregularidade esperada da superfície de busca.

Assim, operadores de mutação que “saltam” para outras regiões longe do ótimo local mais próximo (ou seja, que possuem maior poder de exploração) pode ter mais chances de contribuir para

a construção de um novo bom alinhamento se comparado a operadores que realizam pequenos deslocamentos localmente para melhorar os bons alinhamentos já construídos (ou seja, que possuem maior poder de explotação). Essa desconfiança também é revelada pelo melhor desempenho em experimentos com mutações *random resetting* em detrimento à mutações *creep mutation*.

Nas próximas seções deste capítulo, serão apresentadas as contribuições produzidas por este trabalho, sugestões para trabalhos futuros e um resumo das principais dificuldades encontradas.

7.1 CONTRIBUIÇÕES DESTE TRABALHO

A modelagem baseada no método dialético de otimização apresentada por este trabalho provou ser capaz de tanto melhorar de forma significativa soluções obtidas pelo ClustalW (pelo uso de sementes) como também de obter boas soluções de forma independente. Além desta, outras importantes contribuições realizadas por esse trabalho são:

- A adaptação do método dialético de otimização: do universo contínuo para o discreto;
- A modelagem inédita orientada ao posicionamento das lacunas ao longo das sequências;
- O desenvolvimento de uma função objetivo capaz de incorporar em uma visão matemática aspectos biológicos importantes como a existência de regiões de alta conservação de resíduos como também de uma penalização menos enviesada para mutações de inserções ou exclusões;
- A parametrização do método dialético para o problema de alinhamento múltiplo onde é sugerido uma quantidade reduzida de gerações por período histórico;
- Parametrizações gerais para o método dialético de otimização como a definição de limiares de contradição máxima e mínima e a definição de mecanismos de controle do tamanho da população;

7.2 Dificuldades encontradas

A maior dificuldade na realização deste trabalho foi o problema de escolher qual a melhor quantidade de lacunas necessárias e suficientes para produzir um bom alinhamento. Um agravante deste problema foi a necessidade de realizar experimentos para testar as diversas sugestões para sua solução.

A manipulação da grande diversidade de parâmetros, tanto para o método dialético como do próprio problema do MSA, é outra grande dificuldades vivida durante a realização deste trabalho pois a obtenção do melhor conjunto de parâmetros exige a realização de uma ainda maior quantidade de experimentos.

A realização deste trabalho também foi dificultada pela extensa e multidisciplinar revisão bibliográfica associada ao problema como também às técnicas de otimização utilizadas. Conceitos de filosofia, biologia, computação e matemática precisaram ser revistos ou aprendidos do zero em poucos meses.

7.3 Trabalhos Futuros

Apesar da simplicidade, a modelagem orientada a lacunas demonstrou bons resultados. Porém, é preciso expandir tal modelo para outras abordagens equivalentes. Uma sugestão seria, não mais representar lacunas de forma isolada mas sim blocos inteiros de abertura. Nesta representação, seriam utilizandos dois inteiros para cada bloco aberto (um valor indicando a posição de início do bloco e outro indicando o comprimento do bloco). Tal sugestão pode ser capaz de reduzir custos computacionais para sequências de comprimento elevado pois é capaz de representar uma maior quantidade de lacunas utilizando uma estrutura de dados menor.

Devido ao conjunto de sintético de dados utilizado, uma realização de experimentos com sequências reais e de comprimentos mais elevados é necessária. Além disso, o estudo realizado também necessita da realização de experimentos com sequências de proteínas como também a

utilização de bases e índices de *benchmark* de MSA.

Trabalhos para promover melhorias na função objetivo sugerida como também estudos comparativos com outras funções existentes na literatura também constituem formas de continuação do estudo realizado por este trabalho.

Por fim, as adaptações realizadas na discretização do ODM podem ser revistas e melhoradas com estudos mais profundos.

REFERÊNCIAS

ALTSCHUL, S. F. *et al.* Basic local alignment search tool. *Journal of molecular biology*, v. 215, n. 3, p. 403–10, 5 out. 1990. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/2231712>. Acesso em: 23 jan. 2014.

BÄCK, T. *Evolutionary Algorithms in Theory and Practice Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, UK: Oxford University Press, 1996. p. 314

BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. *Handbook of Evolutionary Computation*. 1st. ed. Bristol, UK: IOP Publishing Ltd, 1997. p. 988

BÄCK, T.; SCHWEFEL, H. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, v. 1, n. 1, p. 1–23, mar. 1993. Disponível em: <http://www.mitpressjournals.org/doi/abs/10.1162/evco.1993.1.1.1>. Acesso em: 16 fev. 2014.

BINGUL, Z.; SEKMEN, A.; ZEIN-SABATTO, S. Evolutionary approach to multi-objective problems using adaptive genetic algorithms. 1923, [S.l.]: IEEE, 1923. p. 1923–1927. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=886394>. Acesso em: 20 fev. 2014.

BIRREN, B. *et al. Pyrenophora tritici-repentis Pt-1C-BFP*. Disponível em: <http://www.ebi.ac.uk/ena/data/view/AAXI01000001-AAXI01000705>. Acesso em: 14 abr. 2013.

BORNHEIM, G. A. *Dialética: teoria, práxis; Ensaio para uma crítica da fundamentação ontológica da Dialética*. São Paulo - SP: Globo-EDUSP, 1977.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: teoria e aplicações*. 1. ed. Rio de Janeiro: Livros Técnicos e Científicos - LTC, 2000. p. 262

BRAMLETTE, M. F. Initialization, mutation and selection methods in genetic algorithms for function optimization. 1991, Los Altos, CA: Morgan Kaufman, 1991. p. 100–107.

CÁNOVAS, F. M. *et al.* Plant proteome analysis. *Proteomics*, v. 4, n. 2, p. 285–98, fev. 2004. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/14760698>. Acesso em: 23 jan. 2014.

CARRILLO, H.; LIPMAN, D. The Multiple Sequence Alignment Problem in Biology. *SIAM Journal on Applied Mathematics*, v. 48, n. 5, p. 1073–1082, out. 1988. Disponível em: <http://epubs.siam.org/doi/abs/10.1137/0148063>. Acesso em: 1 mar. 2014.

CLANCY, S. Chemical structure of RNA. *Nature Education 7(1):60*, v. 7, n. 1, p. 60, 2014. Disponível em: <http://www.nature.com/scitable/topicpage/chemical-structure-of-rna-348>.

CORPET, F. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research*, v. 16, n. 22, p. 10881–10890, 1988. Disponível em: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/16.22.10881>. Acesso em: 14 fev. 2014.

CUTELLO, V. *et al.* Protein multiple sequence alignment by hybrid bio-inspired algorithms. *Nucleic acids research*, v. 39, n. 6, p. 1980–92, mar. 2011. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3064771&tool=pmcentrez&rendertype=abstract>. Acesso em: 31 jul. 2012.

DAUGELAITE, J.; O' DRISCOLL, A.; SLEATOR, R. D. An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics. *ISRN Biomathematics*, v. 2013, p. 1–14, 2013. Disponível em: <http://www.hindawi.com/isrn/biomathematics/2013/615630/>.

DE JONG, K. A. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. 1975. University of Michigan, 1975.

DE SOUZA, R.; DOS SANTOS, W.; YARA, R. Uma abordagem para alinhamento múltiplo de sequências de DNA usando Algoritmos Genéticos e número variável de lacunas. 2013, Fortaleza: ENIAC, 2013. p. 12.

DOS SANTOS, W. *et al.* A dialectical approach for classification of DW-MR Alzheimer's images. jun. 2008, [S.l.]: IEEE, jun. 2008. p. 1728–1735. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4631023>. Acesso em: 20 fev. 2014.

DOS SANTOS, W. *Método Dialético de Busca e Otimização para Análise de Imagens de Ressonância Magnética*. 2009. Universidade Federal de Campina Grande, 2009. Disponível em: <http://www.dominiopublico.gov.br/pesquisa/DetalheObraForm.do?select_action=&co_obra=157400>.

DOS SANTOS, W.; DE ASSIS, F. Optimization based on dialectics. jun. 2009, [S.l.]: IEEE, jun. 2009. p. 2804–2811. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5178738>. Acesso em: 19 fev. 2014.

EBERHART, R.; SHI, Y. *Computational Intelligence: concepts to implementations*. 1. ed. [S.l.]: Morgan Kaufmann, 2007. p. 496

EDGAR, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, v. 32, n. 5, p. 1792–7, jan. 2004. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=390337&tool=pmcentrez&rendertype=abstract>. Acesso em: 22 fev. 2014.

EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. [S.l: s.n.], 2007. p. 299

ELGAR, G.; VAVOURI, T. Tuning in to the signals: noncoding sequence conservation in vertebrate genomes. *Trends in genetics : TIG*, v. 24, n. 7, p. 344–52, jul. 2008. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/18514361>. Acesso em: 24 jan. 2014.

FENG, D. F.; DOOLITTLE, R. F. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of molecular evolution*, v. 25, n. 4, p. 351–60, jan. 1987. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/3118049>.

FERREIRA, R. *Bates, Darwin, Wallace e a Teoria da Evolução*. [S.l.]: UnB-Edusp, 1990.

FOGEL, D. B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. First Edit ed. Piscataway, NJ: IEEE Press, 1995.

FOGEL, L. J.; OWENS, A. J.; WALSH, M. J. *Artificial Intelligence through Simulated Evolution*. Washington DC: John Wiley, 1966.

FRIESEN, T. L.; FARIS, J. D. Molecular mapping of resistance to Pyrenophora tritici-repentis race 5 and sensitivity to Ptr ToxB in wheat. *TAG. Theoretical and applied genetics. Theoretische und angewandte Genetik*, v. 109, n. 3, p. 464–71, ago. 2004. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/15292990>. Acesso em: 14 abr. 2013.

GALDINO, A. S. *et al.* Caracterização molecular de acessos de Cratylia argentea e sua relação filogenética com outras leguminosas. *Pesquisa Agropecuária Brasileira*, v. 45, n. 8, p. 846–854, ago. 2010. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-204X2010000800010&lng=pt&nrm=iso&tlng=pt>. Acesso em: 13 fev. 2014.

GRAMSCI, A. *Introdução ao Estudo da Filosofia e do Materialismo Histórico*. São Paulo - SP: Civilização Brasileira, 1989.

HENIKOFF, S.; HENIKOFF, J. G. Amino acid substitution matrices from protein blocks.

Proceedings of the National Academy of Sciences of the United States of America, v. 89, n. 22, p. 10915–9, 15 nov. 1992. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=50453&tool=pmcentrez&rendertype=abstract>. Acesso em: 20 mar. 2014.

HIGGINS, D. G.; SHARP, P. M. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, v. 73, n. 1, p. 237–44, 15 dez. 1988a. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/3243435>.

HIGGINS, D. G.; SHARP, P. M. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, v. 73, n. 1, p. 237–44, 15 dez. 1988b. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/3243435>.

HOGEWEG, P.; HESPER, B. The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *Journal of Molecular Evolution*, v. 20, n. 2, p. 175–186, jun. 1984. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/6433036>. Acesso em: 14 fev. 2014.

HOLLAND, J. *Adaptation in Natural and Artificial Systems*. 1st Editio ed. Cambridge, MA: MIT Press, 1992.

HOLLAND, J. H. Genetic Algorithms and the Optimal Allocation of Trials. *SIAM Journal on Computing*, v. 2, n. 2, p. 88–105, jun. 1973. Disponível em: <http://epubs.siam.org/doi/abs/10.1137/0202009>. Acesso em: 20 fev. 2014.

HUANG, X.; MILLER, W. A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, v. 12, n. 3, p. 337–357, set. 1991. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/019688589190017D>. Acesso em: 1 mar. 2014.

IBARRA-LACLETTE, E. *et al.* Architecture and evolution of a minute plant genome. *Nature*, v. 498, n. 7452, p. 94–8, 6 jun. 2013. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/23665961>. Acesso em: 22 jan. 2014.

IWAMATSU, M. Generalized evolutionary programming with Lévy-type mutation. *Computer Physics Communications*, v. 147, n. 1-2, p. 729–732, ago. 2002. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0010465502003867>. Acesso em: 1 mar. 2014.

KARP, G. *Cell and molecular biology: concepts and experiments*. [S.l.]: John Wiley, 2008. p. 49–64Disponível em: <http://books.google.com.br/books?id=89BqAAAAMAAJ>.

KATOH, K.; STANDLEY, D. M. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, v. 30, n. 4, p. 772–80, abr. 2013. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3603318&tool=pmcentrez&rendertype=abstract>. Acesso em: 19 fev. 2014.

KEMCCALLUM. *Tradução do DNA em mRNA*. Disponível em: <http://kemccalllum.edublogs.org/2011/06/07/3-5-2-transcription/>. Acesso em: 10 jan. 2013.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, v. 4, p. 1942–1948, 1995. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=488968>.

KONDER, L. *O que é Dialética*. São Paulo - SP: Brasiliense, 2004.

LAMORTE, W. W. *Níveis de Estruturas de Proteínas*. Disponível em: <http://sphweb.bumc.bu.edu/otlt/mph-modules/ph/ph709_a_cellular_world/ph709_a_cellular_world6.html>. Acesso em: 10 jan. 2014.

LARKIN, M. A *et al.* Clustal W and Clustal X version 2.0. *Bioinformatics (Oxford, England)*, v. 23, n. 21, p. 2947–8, 1 nov. 2007. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/17846036>. Acesso em: 22 jan. 2014.

LASSMANN, T.; SONNHAMMER, E. L. . Quality assessment of multiple alignment programs.

FEBS Letters, v. 529, n. 1, p. 126–130, 2 out. 2002. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/12354624>. Acesso em: 20 fev. 2014.

LASSMANN, T.; SONNHAMMER, E. L. L. Kalign--an accurate and fast multiple sequence alignment algorithm. *BMC bioinformatics*, v. 6, p. 298, jan. 2005. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1325270&tool=pmcentrez&rendertype=abstract>. Acesso em: 24 fev. 2014.

LLOYD, S.; SNELL, Q. O. Accelerated large-scale multiple sequence alignment. *BMC bioinformatics*, v. 12, n. 1, p. 466, jan. 2011. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3310909&tool=pmcentrez&rendertype=abstract>. Acesso em: 20 fev. 2014.

LÖWY, M. *Ideologias e Ciências Sociais: elementos para uma análise marxista*. 18 ed. ed. São Paulo - SP: Cortez, 2008. p. 127

MARX, K. *Manuscritos Economico-Filosoficos*. Sao Paulo: Martin Claret, 2002.

MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

NEEDLEMAN, S. B.; WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, v. 48, n. 3, p. 443–453, mar. 1970. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/5420325>. Acesso em: 22 jan. 2014.

NOTREDAME, C. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, v. 3, n. 1, p. 131–44, jan. 2002. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/11966409>.

NOTREDAME, C. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Research*, v. 24, n. 8, p. 1515–1524, 15 abr. 1996. Disponível em: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/24.8.1515>. Acesso em: 20 fev. 2014.

NOTREDAME, C.; HIGGINS, D. G.; HERINGA, J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, v. 302, n. 1, p. 205–17, 8 set. 2000. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/10964570>. Acesso em: 21 jan. 2014.

NOTREDAME, C.; HOLM, L.; HIGGINS, D. G. COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*, v. 14, n. 5, p. 407–422, 1 jun. 1998. Disponível em: <http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/14.5.407>. Acesso em: 25 jan. 2014.

NYRÉN, P. The history of pyrosequencing. *Methods in molecular biology (Clifton, N.J.)*, v. 373, p. 1–14, jan. 2007. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/17185753>. Acesso em: 24 fev. 2014.

PELLINI, C. A. *PLANETA Sedna*. Disponível em: <http://www.portalplanetasedna.com.ar/archivos_varios/mendel01.jpg>. Acesso em: 10 jan. 2014.

RAHNAMAYAN, S.; TIZHOOSH, H. R.; SALAMA, M. M. A. A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, v. 53, n. 10, p. 1605–1614, maio 2007. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0898122107001344>. Acesso em: 1 mar. 2014.

RAMPITSCH, C.; BYKOVA, N. V. Methods for functional proteomic analyses. *Methods in molecular biology (Clifton, N.J.)*, v. 513, p. 93–110, jan. 2009. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/19347646>. Acesso em: 24 fev. 2014.

REINERT, K.; STOYE, J.; WILL, T. An iterative method for faster sum-of-pairs multiple sequence alignment. *Bioinformatics*, v. 16, n. 9, p. 808–814, 1 set. 2000. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/11108703>. Acesso em: 14 fev. 2014.

RONAGHI, M. DNA SEQUENCING:A Sequencing Method Based on Real-Time Pyrophosphate. *Science*, v. 281, n. 5375, p. 363–365, 17 jul. 1998. Disponível em: <http://www.sciencemag.org/cgi/doi/10.1126/science.281.5375.363>. Acesso em: 23 fev. 2014.

RONAGHI, M. *et al.* Real-time DNA sequencing using detection of pyrophosphate release. *Analytical biochemistry*, v. 242, n. 1, p. 84–9, 1 nov. 1996. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/8923969>. Acesso em: 21 jan. 2014.

ROTHAMSTED. *Estruturas Secundárias*. Disponível em: <http://www.rothamsted.ac.uk/notebook/prot.htm>. Acesso em: 10 jan. 2014.

SANGER, F.; COULSON, A. R. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, v. 94, n. 3, p. 441–448, 25 maio 1975. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/1100841>. Acesso em: 16 fev. 2014.

SANTANA, F.; CLEBSCH, C.; FRIESEN, T. *Caracterização de raças de Pyrenophora tritici-repentis, agente etiológico da mancha amarela do trigo, no sul do Brasil.* Passo Fundo, RS: Embrapa Trigo, 2008. Disponível em: <www.infoteca.cnptia.embrapa.br:doc/852141>. Acesso em: 28 fev. 2014.

SCITABLE BY NATURE EDUCATION. *Protein Function*. Disponível em: <http://www.nature.com/scitable/topicpage/protein-function-14123348>. Acesso em: 7 fev. 2014.

SHEN, L.; HE, J. Evolutionary Programming using a mixed strategy with incomplete information. set. 2010, [S.l.]: IEEE, set. 2010. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5625571>. Acesso em: 20 fev. 2014.

SIEVERS, F. *et al.* Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular systems biology*, v. 7, n. 539, p. 539, jan. 2011. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3261699&tool=pmcentrez&rendertype=abstract>. Acesso em: 21 jan. 2014.

SIMOSSIS, V.; KLEINJUNG, J.; HERINGA, J. An overview of multiple sequence alignment. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, v. Chapter 3, p. Unit 3.7, nov. 2003. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/18428699>. Acesso em: 20 fev. 2014.

SÓ BIOLOGIA. Disponível em: <http://www.sobiologia.com.br/conteudos/Citologia2/AcNucleico6.php>. Acesso em: 10 jan. 2014.

STACKEXCHANGE. *Cauchy vs Gaussian*. Disponível em: <http://stats.stackexchange.com/questions/36027/why-does-the-cauchy-distribution-have-no-mean>.

STOYE, J.; MOULTON, V.; DRESS, A. W. M. DCA: An efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Bioinformatics*, v. 13, n. 6, p. 625–626, 1997. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0959440X06000704>. Acesso em: 14 fev. 2014.

TAYLOR, W. R. A flexible method to align large numbers of biological sequences. *Journal of Molecular Evolution*, v. 28, n. 1-2, p. 161–169, dez. 1988. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/3148736>. Acesso em: 14 fev. 2014.

THOMPSON, J. D. *et al.* A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS one*, v. 6, n. 3, p. e18093 (total 14 pages), jan. 2011. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3069049&tool=pmcentrez&rendertype=abstract>. Acesso em: 24 jul. 2012.

THOMPSON, J. D.; PLEWNIAK, F.; POCH, O. A comprehensive comparison of multiple

sequence alignment programs. *Nucleic acids research*, v. 27, n. 13, p. 2682–90, 1 jul. 1999. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=148477&tool=pmcentrez&rendertype=abstract>.

URSEM, R. K. Diversity-Guided Evolutionary Algorithm. 2002, London, UK: Springer-Verlag, 2002. p. 462–474.

VAZQUEZ, A. S. *Filosofia da Praxis*. São Paulo: Expressao Popular, 2007.

VENTER, J. C. GENOMICS: Shotgun Sequencing of the Human Genome. *Science*, v. 280, n. 5369, p. 1540–1542, 5 jun. 1998. Disponível em: <http://www.sciencemag.org/cgi/doi/10.1126/science.280.5369.1540>. Acesso em: 24 fev. 2014.

WANG, X. *et al.* Particle Swarm Optimization with Escape Velocity. nov. 2006, [S.l.]: IEEE, nov. 2006. p. 457–460. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4072129>. Acesso em: 20 fev. 2014.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and Computing*, v. 4, n. 2, jun. 1994. Disponível em: <http://link.springer.com/10.1007/BF00175354>. Acesso em: 20 fev. 2014.

YAO, X.; LIU, Y.; LIN, G. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 2, p. 82–102, jul. 1999. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771163>. Acesso em: 3 fev. 2014.

ZAVALA, A. E. M.; AGUIRRE, A. H.; DIHARCE, E. R. V. Particle Evolutionary Swarm Optimization Algorithm (PESO). 2005, [S.l.]: IEEE, 2005. p. 282–289. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1592230>. Acesso em: 20 fev. 2014.

ZHU, C.; JIANG, T. Multicontext fuzzy clustering for separation of brain tissues in magnetic resonance images. *NeuroImage*, v. 18, n. 3, p. 685–96, mar. 2003. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/12667846>. Acesso em: 1 mar. 2014.

APÊNDICE A:

Uma abordagem para alinhamento múltiplo de sequências de DNA usando Algoritmos Genéticos e número variável de lacunas

Trabalho apresentado no X Encontro Nacional de Inteligência Artificial e Computacional

Uma abordagem para alinhamento múltiplo de sequências de DNA usando Algoritmos Genéticos e número variável de lacunas

Rodrigo G. de Souza, Ricardo Yara, Wellington P. dos Santos

Departamento de Engenharia Biomédica – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

rodrigogs2@gmail.com, ricardo.yara@gmail.com, wellington.santos@ieee.org

Abstract. *This paper presents an approach based on genetic algorithms to solve the problem of Multiple Sequence Alignment (MSA) of DNA sequences. In our approach, MSA problems are viewed as optimization problems, where the objective function is the overall matching score and the solution candidates are modeled as vectors composed by the position of gaps and by the amount of gaps, for each sequence. In order to test our proposal, we constructed a database composed by 50 group of 4 sequences of 60 bases of nucleotides each one. In comparison with the classic Clustal algorithm, our proposal proved to be able to make significant improvements in the Clustal alignments and also to produce high scoring alignments independently.*

Resumo. *Este artigo apresenta uma abordagem baseada em algoritmos genéticos para resolver o problema do alinhamento múltiplo de sequências (MSA) de DNA. Em nossa abordagem, problemas de múltiplo alinhamento de sequências são vistos como problemas de otimização, onde a função objetivo é função de pontuação geral de correspondências e os candidatos à solução são modelados como vetores compostos pela posição de lacunas e pela quantidade de lacunas, para cada sequência. A fim de testar a nossa proposta, construímos uma base de dados composta por 50 grupos de 4 sequências de 60 bases de nucleótidos cada uma. Em comparação com o algoritmo Clustal clássico, a nossa proposta provou ser capaz de realizar melhorias significativas nos alinhamentos realizados pelo Clustal como também de produzir alinhamentos de altos scores de forma independente.*

1. Introdução

O alinhamento múltiplo de sequências (*Multiple Sequence Alignment*, MSA) é um dos problemas candentes da Bioinformática, tanto do ponto de vista das complexidades computacional e biológica envolvidas na construção dos métodos de alinhamento quanto em relação à importância da pesquisa genética para o contexto atual. A aplicação de tais métodos a sequências de nucleotídeos pode ser capaz de encontrar regiões de alta similaridade que podem ser comum a uma família e de identificar novos membros de uma família, desde que seja definida uma função objetivo adequada [1][2][3].

O desenvolvimento de métodos iterativos de alinhamento múltiplo de sequências através da investigação de técnicas de modelagem evolucionária têm disponibilizado

ferramentas mais adequadas a situações específicas tornando possível assim que os pesquisadores obtenham bons alinhamentos que atendam às mais variadas exigências e restrições [2][3]. Além disso, a característica iterativa de tais métodos pode ser usada para melhorar resultados obtidos por alinhamentos produzidos por outros métodos bem estabelecidos.

As variações observadas do produto de um alinhamento múltiplo de sequências podem ser consideradas consequências da ocorrência de três tipos principais de eventos biológicos: a inserção ou surgimento de bases, a exclusão de bases ou a mutação de bases.

Em uma visão orientada a lacunas (*gaps*), uma abstração de um alinhamento múltiplo de sequências pode ser definida como a dispersão de lacunas ao longo das sequências envolvidas. Este trabalho faz uso dessa abstração para definir a representação para um MSA na qual um alinhamento é visto como um mapeamento da distribuição das lacunas entre as sequências, isto é, um alinhamento é modelado como uma lista das posições onde existam lacunas em cada uma das sequências relacionadas.

Esta representação torna possível que a busca pelo melhor alinhamento seja vista como um problema de otimização, permitindo assim o uso de métodos de otimização iterativos, como algorítimos genéticos. A modelagem sugerida também permite que soluções produzidas por outros métodos sejam inseridas na população inicial.

Este trabalho está organizado como segue: na seção 2 são apresentadas a base de dados e as sequências utilizadas no estudo de alinhamento múltiplo de sequências. Na mesma seção também é proposta uma nova função objetivo para alinhamento múltiplo baseada na modelagem dos candidatos ao melhor alinhamento na forma de vetores discretos contendo as posições das lacunas a serem inseridas, permitindo que seja utilizada um número variável de lacunas. Também são apresentadas as diversas abordagens que orientaram a efetivação dos experimentos, a saber: geração de resultados com algoritmos genéticos e a função objetivo proposta em contraposição aos resultados do método Clustal e melhoramento dos resultados do Clustal a partir da adaptação da proposta deste trabalho. Na seção 3 são apresentados e comentados os resultados dos experimentos. Por fim, na seção 4 são apresentadas discussões e conclusões.

2. Materiais e Métodos

2.1. Sequências estudadas

O fungo *Phyrenophora tritici-repentis* [4] é o agente patogênico da mancha amarela, chamada de *tan spot*, uma das doenças de maior impacto econômico que afetam o trigo no Brasil como também em todo o mundo [5][6]. Quando expostas às condições favoráveis, lavouras infectadas por esta patologia foliar podem ter suas produtividades de grãos reduzidas drasticamente principalmente pelos severos danos provocados ao principal mecanismo autotrófico do vegetal atacado, a fotossíntese [6].

Múltiplos tipos deste patogênico têm sido caracterizados através de suas habilidades de causar necrose e/ou cloroses em diferentes tipos de trigo. Tais diferenças sintomáticas entre variadas espécies de trigo têm motivado diversos trabalhos de investigação sobre a resistência específica de algumas espécies de trigos à clorose ou à necrose desenvolvida.

No início dos anos 2000, pesquisas buscaram identificar a existência de determinantes genéticos nos trigos para tais resistências. Já em estudos mais recentes [7] [8][9] realizados através de comparações de análises genômicas de diferentes fungos patogênicos revelam a ocorrência de transferência horizontal de genes malignos entre tais agentes patogênicos. Tais estudos defendem que próprio *Phyrenophora tritici-repentis* adquiriu genes malignos de outra espécie de patogênico, o *Phaeosphaeria nodorum*.

Portanto, faz-se necessário o estudo e processamento de informações genéticas do *Phyrenophora tritici-repentis* para que, através dos dados obtidos sobre a localização e identificação de genes malignos, nos fungos, e dos determinantes genéticos à resistência natural, em variadas espécies de trigo envolvidas, possam ser desenvolvidas novas lavouras mais resistentes.

2.2. Base de dados EBI

O banco de dados eleito para a importação das sequências escolhidas foi o EBI, Instituto de Bioinformática Europeu – um centro de pequisa e serviços em bioinformática que faz parte do Laboratório Europeu de Biologia Molecular, EMBL – pois trata-se de um dos pioneiros na disponibilização de ferramentas e estabelecimento de padrões para a troca de dados biológicos entre pesquisadores de todo o mundo.

2.3. Algoritmos genéticos, modelagem dos candidatos à melhor solução e função objetivo proposta

Assim como as demais classes de algoritmos evolucionários, para responder de forma desejada os algoritmos genéticos necessitam de uma modelagem que defina um correto mapeamento para o genótipo, através da modelagem da representação, e para a avaliação do fenótipo, através da definição da função objetivo [10].

Neste trabalho é proposta uma nova função objetivo (OF) para alinhamento múltiplo de sequências, adequada à construção de métodos de alinhamento baseados em meta-heurísticas da Computação Evolucionária e sensivelmente diferente das funções objetivo utilizadas em outros métodos como em [11] e [12].

A modelagem usada por este trabalho para candidatos à solução trouxe um problema relacionado à definição da quantidade necessária de lacunas para a realização de um bom alinhamento. Tal questão foi solucionada permitindo-se que as posições das lacunas assumam valores repetidos. Dessa forma, a evolução natural dos candidatos à solução é responsável por escolher também a quantidade de lacunas necessária. Essa modelagem também permite que diferentes quantidades de lacunas possam ser utilizadas para cada uma das sequências e essa interessante característica não é observada em outros métodos de alinhamento [11] e [12]. Porém, foi necessário estabelecer um parâmetro com o propósito de definir o número máximo de lacunas que podem ser inseridas em uma sequência relativo ao comprimento desta.

Os operadores de variação utilizados foram o *uniform crossover,* como operador de cruzamento, *random resetting* e o *creep mutation*, ambos como operador de mutação. O operador *creep mutation* adaptado para uma versão na qual a pequena variação do alelo é extraída seguindo uma distribuição normal.

Tabela 1. Valores utilizados para as principais variáveis do GA e do MSA

Prob. de Cruzamento	0,75
Prob. de Mutação	0,2
Número Máximo de Gerações	1500
Tamanho da População	120
Tamanho do Conjunto Elite	10% da População
Comprimento das sequências (bases)	60
Número Máximo de Lacunas (%)	50% do comprimento das sequências

A seleção de pais foi realizada em torneios com tamanho igual a 20 indivíduos enquanto que na seleção de sobrevivência, do tipo $\mu+\mu$, todos os μ indivíduos que foram candidatos a pais são descartados ao final de cada geração. Para preservar os indivíduos mais aptos a técnica elitismo foi usada. Os valores dos principais parâmetros utilizados são listados na Tabela 1. Tais valores permanecem inalterados para todas abordagens.

2.4. Modelagem matemática

Neste trabalho um alinhamento de sequências é representado através de um vetor de posições onde foram inseridas lacunas nas sequências envolvidas. Baseada nesta representação, uma definição formal de um indivíduo, o qual representa um alinhamento candidato à solução do problema, é sugerida como se segue:

$w_i = \{ g_{0,0} , g_{0,1} , \dots , g_{0,M} , g_{1,0} , g_{1,1} , \dots , g_{1,M} , \dots , g_{N,0} , g_{N,1} , \dots , g_{N,M} \}$, onde:

N corresponde ao número de sequências do alinhamento

M corresponde ao total de lacunas (*gaps*) existentes em cada sequência

P corresponde ao total de indivíduos existentes na população

L corresponde ao comprimento máximo de cada sequência

$g_{n,m}$ é um número inteiro o qual corresponde à posição da m-ésima lacuna na n-ésima sequência com $n \in \{0,1,2,...,N\}$ e $m \in \{0,1,2,...,M\}$

w_i corresponde ao i-ésimo indivíduo da população com $i \in \{0,1,2,\dots,P\}$

2.5. Modelagem da Função objetivo

Neste trabalho, a modelagem da função objetivo é baseada na construção de uma função *matching score* a qual é realizada sob a ideia de premiar características desejáveis e penalizar características não desejadas para um dado alinhamento. Isso define bem o caráter de otimização através da maximização da função *matching score* a qual penaliza inserções e maus posicionamentos de *gaps* como também soma pontos extras ao acumular equivalências em uma mesma coluna e ao detectar similaridades em colunas vizinhas.

Assim como outros métodos de alinhamento já conhecidos, a função *matching score* para esse trabalho também faz uso de uma matriz de similaridade para pontuar os *matches* e *mismatches* obtidos durante a análise de similaridade de cada coluna [13][14] [15]. Na OF proposta, é possível reforçar uma preferência à ocorrência de *matches* nas mesmas colunas em detrimento aqueles que apareçam em colunas diferentes e de forma isolada. Tal recurso permite que elevados índices de conservação entre as sequências sejam detectados durante o processo de construção do alinhamento. De maneira

equivalente, também é possível premiar alinhamentos cujas colunas de elevada conservação estejam mais próximas uma das outras para que, dessa forma, sejam detectadas regiões de alta similaridade. Tais preferências são justificadas por reconhecer que regiões bem alinhadas são mais significantes biologicamente [16], pois trazem evidências mais fortes da existência de homologia entre as sequências.

Portanto, a função *matching score* desenvolvida leva em conta quatro aspectos ao avaliar um alinhamento: (1) o cálculo do somatório produzido pela comparação, em cada coluna, da primeira base com as demais bases, utilizando a matriz de similaridade, (2) a quantidade de *matches* encontrados em uma mesma coluna, (3) a proximidade entre colunas que possuam uma grande quantidade de *matches*, (4) a abertura de *gaps*, sua concentração em uma mesma coluna como também em uma mesma linha, porém para esta última se leva em conta também a consecutividade.

A avaliação do alinhamento é vista como uma função multiobjetivo e, portanto, definida pela composição de quatro funções onde cada uma representa um dos aspectos anteriormente citados. Para os aspectos (1), (2) e (3), os quais tratam características desejadas, serão associados a funções pontuação positiva enquanto que para o aspecto (4) será associada uma função de pontuação negativa pois tal aspecto trata de características que devem ser evitadas ou minimizadas. A combinação dessas subfunções evidencia o caráter de otimização configurando assim o problema exposto como um problema de maximização.

Assim, sejam $Col_c(w_i)$, com $c \in \{1,2,...,C)\}$ onde $C=T+L$, e $Line_n(w_i)$ com $n \in \{1,2,...,N)\}$ respectivamente a c-ésima coluna e a n-ésima sequência de um alinhamento representado pelo indivíduo w_i, a função objetivo f aplicada a w_i é definida como:

$$f(w_i) = f_{gaps}(w_i) + f_{matches}(w_i)$$

onde

$$f_{gaps}(w_i) = \rho_{line} \cdot \sum_{n=1}^{N} f_{penLine}((w_i)) + \rho_{column} \cdot \sum_{n=1}^{N} f_{penColumn}(Col_c(w_i)),$$

$$f_{matches}(w_i) = \sum_{c=1}^{C} \{\beta_{numMatches} \cdot f_{numDeMatches}[Col_c(w_i)]\} + \{\beta_{consec} \cdot f_{consec}[Col_c(w_i)]\} + f_{matrixIUB}[Col_c(w_i)]$$

$$e$$

$$\rho_{line}, \rho_{column}, \beta_{numMatches}, \beta_{consec} \in [0,1]$$

2.6. Construção e formatação dos dados

Na construção da base de dados para esse trabalho foram utilizadas as sequências *AAXI01000001* até *AAXI01000040* da base pública do EMBL-EB. Elas correspondem a sequenciamentos para o genoma do fungo *Pyrenophora tritici-repentis* Pt-1C-BFP [4].

Para cada um destes sequenciamentos foi capturadas apenas as primeiras 960 bases que foram distribuídas em linhas de comprimento igual a 60 bases. Para a construção de cada um conjuntos de sequências, foram escolhidos uniformemente ao acaso quatro desses genomas e, de cada um, foi extraída a *n*-ésima linha onde n corresponde a um número aleatório entre 1 e 16 com distribuição uniforme. Uma vez identificadas, estas quatro linhas irão compor um conjunto de quatro sequências o qual será usado exclusivamente por um dos ensaios. No total foram construídos 50 conjuntos

de 4 sequências cada.

Após serem definidos, cada conjunto de sequências é alinhado através método de Clustal W através da ferramenta Clustal W2 disponível no *website* do EMBL-EBI. Para todos alinhamentos foram utilizados os parâmetros padrões da ferramenta, conforme a Tabela 2. Estes alinhamentos são salvos para posteriormente (1) serem avaliados pela mesma função objetivo que é aplicada a todos os alinhamentos encontrados pelo GA e (2) serem usados para semear a população inicial do GA em alguns experimentos.

Tabela 2. Valores utilizados para os parâmetros da ferramenta Clustal

parâmetro	Descrição	valor usado
dnamatrix	DNA/RNA scoring matrix	IUB
gapopen	gap creation penalty	-10
noendgaps	no end gap separation penalty	no
gapext	gap extension penalty	-0,2
gapdist	gap separation penalty	5
iteration	iteration strategy	none
numiter	maximum number of iterations	1

2.7. Experimentos

Os experimentos foram realizados ao longo de 50 ensaios e, para cada um deles, foi utilizado um determinado conjunto de sequências obtido durante a seleção e construção da base de dados. Em cada ensaio, o conjunto de sequências é alinhado através de três diferentes abordagens e os scores obtidos são comparados entre si e com aquele obtido pelo alinhamento sugerido pelo Clustal. As abordagens sugeridas são: (1) algoritmo genético tradicional para (GA Canônico), (2) algoritmo genético usando o número de lacunas *Clustal* e (3) GA com número de lacunas flutuante.

Para cada um desses ensaios, foram realizados dez execuções (experimentos) do algoritmo genético. Os valores dos parâmetros específicos ao algoritmo genético foram os mesmos para todas as abordagens, enquanto que o parâmetro de número máximo de lacunas teve sua interpretação variando conforme cada abordagem.

A cada uma das abordagens, além da versão prevista, foi também permitida a execução de uma variação do algoritmo a qual faz uso da técnica de semeação da população. A semente utilizada corresponde ao alinhamento sugerido pelo método Clustal W. Na criação do indivíduo semente pode ocorrer que a quantidade de lacunas usadas pelo alinhamento Clustal seja menor que o número máximo de lacunas disponível pela modelagem. Nesse caso, as posições não utilizadas recebem um valor que não produz alterações no alinhamento (por exemplo valores repetidos), porém nada impede que esse valor seja modificado durante a evolução. A utilização de sementes visa a possibilidade de promover melhorias nos alinhamentos realizados por uma ferramenta clássica de uso já estabelecido, como o Clustal, além de permitir o uso de outros recursos, como a construção da árvore filogenética, providos por tais ferramentas.

2.8. GA Canônico

Nessa abordagem, o algoritmo genético, quando utilizado sem variações, é executado sem fazer uso de nenhuma informação prévia sobre os alinhamentos sugeridos pelo

Clustal. Para todos os indivíduos da população inicial, todas as posições de lacuna têm seus valores aleatoriamente escolhidos (repetições de valores são permitidas neste processo). Assim, o total de posições geradas ao acaso é igual ao número de lacunas máximo. Naturalmente a quantidade de lacunas efetivamente inseridas em cada sequência poderá variar conforme o andamento do algoritmo durante a evolução.

Na variação dessa abordagem, ocorre a inserção de um indivíduo semente à população inicial. Portanto, todos indivíduos da população inicial terão todas as posições de lacunas iniciadas aleatoriamente obedecendo ao parâmetro **número máximo de lacunas**, com exceção do indivíduo semente o qual terá uma quantidade de posições de lacunas iniciadas orientada pelo alinhamento correspondente à semente.

2.9. GA com o mesmo Número de Lacunas utilizado pelo Clustal

Nesta modelagem, a solução sugerida pelo Clustal governa a quantidade de posições de lacunas cujos valores foram inicializados. Isso pode ajudar na convergência do algoritmo para os casos onde o número de lacunas necessário para a construção de um alinhamento satisfatório seja próximo ao número usado efetivamente pelo alinhamento realizado pelo Clustal W.

Na variação dessa abordagem, em cada ensaio, todos os indivíduos da população (inclusive a semente) têm a mesma quantidade de posições de lacunas inicializadas – assim como na abordagem original.

2.10. GA Canônico com Número de Lacunas Flutuante

Intuitivamente pode-se afirmar que o número de lacunas utilizado no alinhamento de um conjunto de sequências acompanhe, de forma inversa, a variação do grau de similaridade existente entre tais sequências.

Através da inspeção dos alinhamentos sugeridos pelo Clustal para os conjuntos de sequências utilizados neste trabalho, percebe-se uma variação superior a 300% no número de lacunas utilizado – mesmo todas as sequências envolvidas possuindo o mesmo comprimento. Diante disso, também seria esperado que usar uma grande quantidade de lacunas além da necessária fatalmente aumentaria a dificuldade de adaptação do algoritmo e, assim, sua convergência.

Portanto, a escolha da quantidade de lacunas é um ponto crítico na busca de soluções satisfatórias. Assim, a adoção de uma abordagem com número flutuante de lacunas cujos valores são efetivamente inicializados faz-se necessária pois é capaz de aumentar a diversidade da população sob o aspecto da quantidade de alterações necessárias a fim de obter-se um alinhamento que necessite de uma determinada quantidade de lacunas.

3. Resultados

A apresentação dos resultados de cada abordagem é estruturada nas Tabelas 3, 4 e 5 nas quais são expressos os *scores* obtidos em todos os ensaios pelo Clustal, pelo GA e pela variação do GA usando sementes para todas as abordagens. A significância estatística dos resultados foi aferida através do teste não-paramétrico de Wilcoxon para variáveis dependentes através do suplemento para o MS Excel chamado Action, o qual se trata de uma extensão para o ambiente estatístico *R*. Na Figura 1 é mostrado um gráfico de linhas que ilustra o *score* obtido por cada abordagem e em cada variação para cada um dos 50 ensaios.

Figura 1. Gráfico dos resultados de todas 3 abordagens e suas variações com uso de sementes.

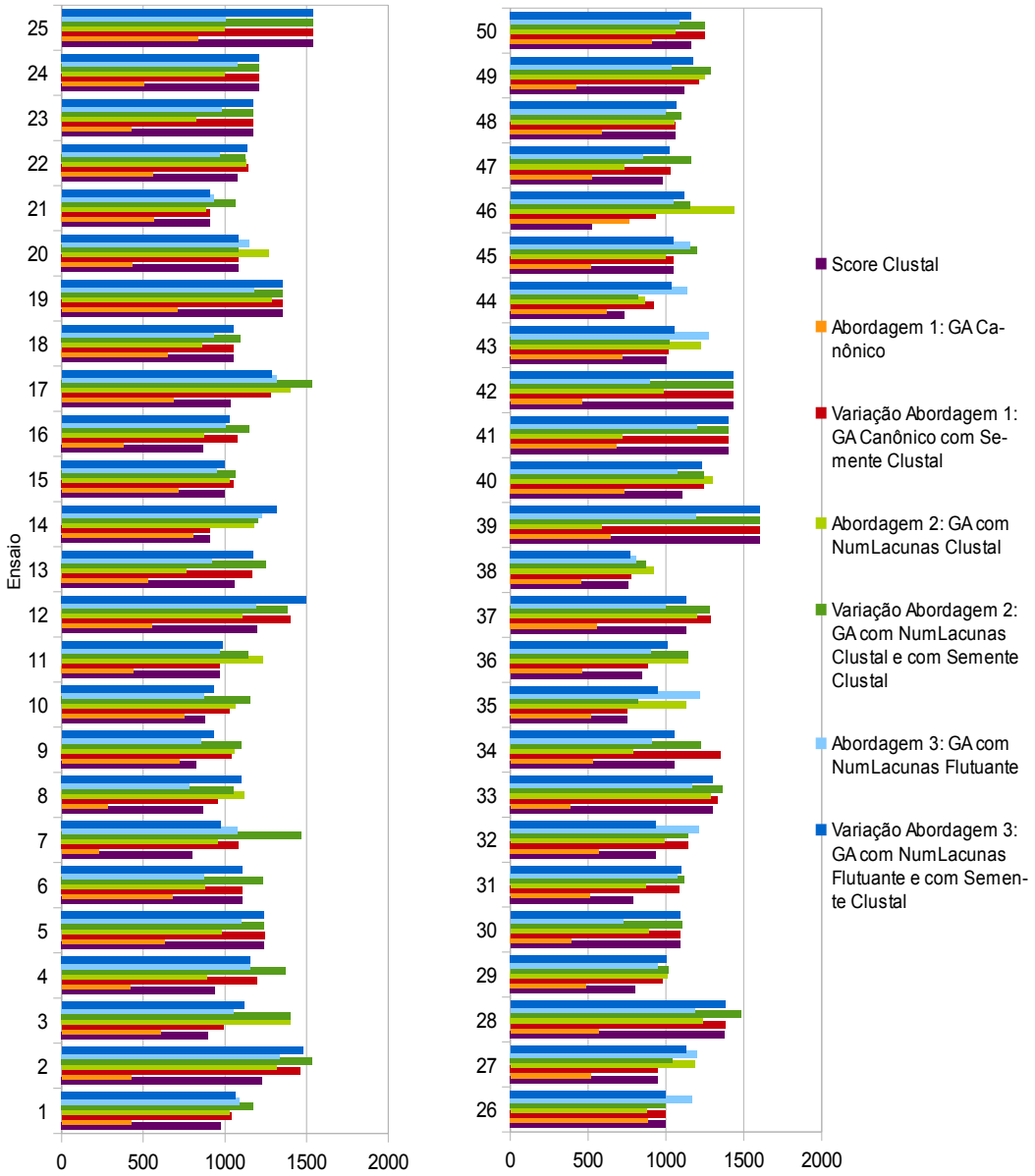

Tabela 3. Scores de todas as abordagens

	Abordagem original							Variação da Abordagem: Uso de Sementes						
		Diferença quando GA venceu			Diferença quando Clustal venceu				Diferença quando GA venceu			Diferença quando Clustal venceu		
Abordagem usada	Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal	Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score GA	Absoluta	Casos	Percentual % entre Diferença Absoluta e Score Clustal
GA Canônico	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	567,24	4,80	1	0,63%	480,32	49	44,31%	1132,14	89,38	32	8,00%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	912,51	239,96	**2%**	31,42%	972,22	**98%**	71,64%	1604,63	412,40	**64%**	44,05%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	226,79	0,00		0,00%	0,00		0,00%	754,97	0,00		0,00%	0,00		0,00%
	Desvio	Desvio		Desvio	Desvio		Desvio	Desvio	Desvio		Desvio	Desvio		Desvio
	Padrão	Padrão		Padrão	Padrão		Padrão	Padrão	Padrão		Padrão	Padrão		Padrão
	149,12	33,94		4,44%	232,59		17,16%	190,94	107,86		9,89%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,9999999994 (H1: clustal menor que GA)							**Teste Wilcoxon (valor p): 4,17144456013841E-007** (H1: clustal menor que GA)						
	Teste Wilcoxon (valor p): 1,1901468699449E-009 (H1: clustal diferente de GA)							**Teste Wilcoxon (valor p): 8,34288912027683E-007** (H1: clustal diferente de GA)						
GA Num Lacunas Clustal	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1045,47	114,63	27	9,60%	111,92	23	8,80%	1207,60	164,84	40	13,49%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1442,70	918,98	**54%**	63,70%	1016,58	**46%**	63,35%	1604,63	669,54	**80%**	54,79%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	588,06	0,00		0,00%	0,00		0,00%	817,65	0,00		0,00%	0,00		0,00%
	Desvio	Desvio		Desvio	Desvio		Desvio	Desvio	Desvio		Desvio	Desvio		Desvio
	Padrão	Padrão		Padrão	Padrão		Padrão	Padrão	Padrão		Padrão	Padrão		Padrão
	195,18	171,49		13,01%	202,23		14,25%	180,46	164,20		12,51%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,3425779004 (H1: clustal menor que GA)							**Teste Wilcoxon (valor p): 1,3555674748993E-008** (H1: clustal menor que GA)						
	Teste Wilcoxon (valor p): 0,6851558008 (H1: clustal diferente de GA)							**Teste Wilcoxon (valor p): 2,71113494979 86E-008** (H1: clustal diferente de GA)						
GA Num Lacunas Flutuante	Média	Média	Total	Média	Média	Total	Média	Média	Média	Total	Média	Média	Total	Média
	1048,71	94,55	24	8,35%	88,60	26	7,10%	1140,95	98,20	29	8,68%	0,00	0	0,00%
	Máximo	Máximo	%	Máximo	Máximo	%	Máximo	Máximo	Máximo	%	Máximo	Máximo	%	Máximo
	1336,44	520,79	**48%**	49,86%	539,65	**52%**	37,66%	1604,63	592,96	**58%**	53,10%	0,00	**0%**	0,00%
	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo	Mínimo	Mínimo		Mínimo	Mínimo		Mínimo
	724,41	0,00		0,00%	0,00		0,00%	773,49	0,00		0,00%	0,00		0,00%
	Desvio	Desvio		Desvio	Desvio		Desvio	Desvio	Desvio		Desvio	Desvio		Desvio
	Padrão	Padrão		Padrão	Padrão		Padrão	Padrão	Padrão		Padrão	Padrão		Padrão
	144,23	138,26		12,11%	133,30		9,79%	177,73	131,61		11,41%	0,00		0,00%
	Teste Wilcoxon (valor p): 0,4538899342 (H1: clustal menor que GA)							**Teste Wilcoxon (valor p): 1,35118457023899E-006** (H1: clustal menor que GA)						
	Teste Wilcoxon (valor p): 0,9077798683 (H1: clustal diferente de GA)							**Teste Wilcoxon (valor p): 2,70236914047798E-006** (H1: clustal diferente de GA)						

4. Discussão e conclusões

A função objetivo proposta neste trabalho, baseada na modelagem dos candidatos ao melhor alinhamento como vetores discretos de posições de lacunas, explorada por meio de um método de alinhamento baseado no algoritmo genético canônico, mostrou-se eficaz quando comparados os seus resultados com aqueles obtidos com uso do Clustal puro, método do estado da arte de alinhamento múltiplo de sequências.

O uso pelo método proposto tanto de informações sobre a quantidade de lacunas utilizadas bem como do próprio alinhamento obtido pelo Clustal, através da semeatura da população inicial, proporcionou oportunidades para o algoritmo genético melhorar de forma significativa os alinhamentos encontrados pelo método Clustal, oferecendo assim uma busca por outros bons alinhamentos, ao mesmo tempo que permite a utilização de outros importantes recursos oferecidos por esse consolidado método como a árvore filogenética.

Além da capacidade de melhorar significativamente os resultados do Clustal, o método sugerido por este trabalho também é capaz de obter alinhamentos de *scores* elevados de forma independente, uma vez que a abordagem proposta conseguiu ser melhor em até 54% dos casos.

Como trabalho futuro, vem sendo estudada a possibilidade de utilização de outros métodos de otimização baseados em meta-heurísticas da Computação Evolucionária diferentes, tais como o método de otimização por enxame de partículas, as estratégias evolutivas, a programação evolucionária, e métodos mais recentes, como o método dialético, como forma de ampliar a investigação a cerca das possibilidades de obtenção de alinhamentos melhores, tanto do ponto de vista objetivo, pelas medidas dos *scores*, quanto do ponto de vista subjetivo, levando-se em conta o conhecimento especialista e o contexto genético dos alinhamentos.

Referências

[1] Mount DM. (2004) “Bioinformatics: Sequence and Genome Analysis”, 2nd ed., Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY. ISBN 0-87969-608-7.

[2] Notredame, Cédric “Recent progresses in multiple sequence alignment: a survey” Pharmacogenomics (2002) 3 (1)

[3] Prakash Lingam K M, Chandrakala S. “A survey on recent developments in multiple sequence alignment methods” J Nat Sc Biol Med 2011;2:96-7

[4] Birren B., Lander E., Galagan J., Nusbaum C., Devon K., Ma L.-J., Jaffe D., Butler J., Alvarez P., Gnerre S., Grabherr M., Kleber M., Mauceli E., Brockman W., MacCallum I.A., Young S., LaButti K., DeCaprio D., Crawford M., Koehrsen M., Engels R., Montgomery P., Pearson M., Howarth C., Larson L., White J., Yandava C., Kodira C., Zeng Q., O'Leary S., Alvarado L., Ciuffetti L. and Pandelova I. (2007), Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA.

[5] Santana, F. M., Clebsch, C. C. and Friesen, T. L. (2008) “Caracterização de raças de Pyrenophora tritici-repentis, agente etiológico da mancha amarela do trigo, no sul do Brasil”, Passo Fundo: Embrapa Trigo,. 13 p. html (Embrapa Trigo. Boletim de Pesquisa e Desenvolvimento Online, 60).

[6] Friesen, T. L., and Faris, J. D. (2004) “Molecular mapping of resistance to Pyrenophora tritici-repentis race 5 and sensitivity to Ptr ToxB in wheat”, TAG. Theoretical and applied genetics, Theoretische und angewandte Genetik, 109(3), 464–71.

[7] Gardiner DM, McDonald MC, Covarelli L, Solomon PS, Rusu AG, et al. (2012) “Comparative Pathogenomics Reveals Horizontally Acquired Novel Virulence Genes in Fungi Infecting Cereal Hosts”, PLoS Pathog 8(9): e1002952.

[8] Friesen TL, Stukenbrock EH, Liu Z, Meinhardt S, Ling H, et al. (2006) “Emergence of a new disease as a result of interspecific virulence gene transfer”, Nat Genet 38: 953–956.

[9] Ma L-J, van der Does HC, Borkovich KA, Coleman JJ, Daboussi M-J, et al. (2010) “Comparative genomics reveals mobile pathogenicity chromosomes in Fusarium”,

Nature 464: 367–373.

[10] Eiben, Agoston E. and Smith, J.E (2007) “Introduction to Evolutionary Computing”, Springer, Natural Computing Series 1st edition, ISBN: 3-540-40184-9.

[11] Gondro, C., and B. P. Kinghorn. “A simple genetic algorithm for multiple sequence alignment”. Genet. Mol. Res 6.4 (2007): 964-982.

[12] C Notredame, L Holm, and D G Higgins, “COFFEE: an objective function for multiple sequence alignments”. Bioinformatics (1998) 14 (5): 407-422.

[13] Larkin M.A., Blackshields G., Brown N.P., Chenna R., McGettigan P.A., McWilliam H., Valentin F., Wallace I.M., Wilm A., Lopez R., Thompson J.D., Gibson T.J. and Higgins D.G. (2007), “Clustal W and Clustal X version 2.0”, Bioinformatics 23(21): 2947-8.

[14] Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990) “Basic local alignment search tool”, Journal of Molecular Biology 215 (3): 403–410.

[15] Notredame, C., Higgins, D.G. (1996) “SAGA: sequence alignment by genetic algorithm”, Nucleic Acids Res 24 (8): 1515–24.

[16] Smith, Temple F. and Waterman, Michael S. (1981). “Identification of Common Molecular Subsequences”. Journal of Molecular Biology 147: 195–197.