UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Anderson Rici Amorim

Alinhamento múltiplo de sequências utilizando algoritmo genético multifunção e colônia de formigas

São José do Rio Preto
2017

Anderson Rici Amorim

Alinhamento múltiplo de sequências utilizando algoritmo genético multifunção e colônia de formigas

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador:

Prof. Dr. Geraldo Francisco Donegá Zafalon

São José do Rio Preto
2017

Amorim, Anderson Rici.
Alinhamento múltiplo de sequências utilizando algoritmo genético multifunção e colônia de formigas / Anderson Rici Amorim. -- São José do Rio Preto, 2017
73 f. : il., tabs.

Orientador: Geraldo Francisco Donegá Zafalon
Dissertação (mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Biociências, Letras e Ciências Exatas

1. Computação - Matemática. 2. Bioinformática. 3. Alinhamento de sequências (Bioinformática) 4. Algoritmos genéticos.
5. Programação heurística. I. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. II. Título.

CDU – 574:518.72

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Anderson Rici Amorim

Alinhamento múltiplo de sequências utilizando algoritmo genético multifunção e colônia de formigas

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Prof. Dr. Geraldo Francisco Donegá Zafalon — Anderson Rici Amorim

Banca Avaliadora:
Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

Profa. Dra. Rogéria Cristiane Gratão de Souza

São José do Rio Preto
2017

Aos meus pais, Dimas e Sandra

À minha avó Elena

À minha amada Jacqueline

À Elizete e a José Carlos

"Sonhe como se fosse viver para sempre, viva como se fosse morrer hoje."

-James Dean

Agradecimentos

Agradeço primeiramente a Deus pelos pilares que sustentam a minha vida, por sempre me alimentar com fé para seguir o caminho que escolhi.

Agradeço infinitamente aos meus pais, Dimas e Sandra, por todo o amor, carinho e suporte, por sacrificar suas vidas para que a minha pudesse seguir um curso que a deles não puderam. Imensa gratidão também à minha avó Elena e ao meu tio Sandro, que me acompanharam desde os meus primeiros passos até agora, sempre sem medir esforços para ajudar-me, não importa como.

Agradeço à minha amada Jacqueline por todo amor, carinho e suporte, não somente nos bons momentos, mas também nos ruins, como deve ser, até o fim. Obrigado pelos sorrisos, lágrimas, por fazer parte de toda a minha caminhada.

Agradeço aos meus amigos Alexandre, Rodrigo, Samuel e Victor, por todos os momentos, por me acompanharem e me apoiarem desde minha infância. A parcela de contribuição de vocês é inestimável. Obrigado!

Agradeço aos colegas de turma que viraram verdadeiros amigos de caminhada na vida acadêmica, Allan e Guilherme. Obrigado por tornarem as coisas mais fáceis, pelas risadas e pelos bons momentos que compartilhamos.

Agradeço infinitamente ao meu orientador, amigo e parceiro de Tênis, Prof. Dr. Geraldo Francisco Donegá Zafalon, pelas inestimáveis contribuições e apoio ao meu trabalho, pelos bons conselhos e por acreditar em meu potencial. Muito obrigado!

Agradeço ao Prof. Dr. Carlos Roberto Valêncio, por inicialmente me acolher junto ao Programa de Pós Graduação em Ciência da Computação da UNESP.

Agradeço a todos os docentes do Programa e aos funcionários do Instituto de Biociências, Letras e Ciências Exatas pelas incontáveis contribuições e esforço despendido.

Agradeço à CAPES pelo fomento à pesquisa associada ao presente trabalho.

Resumo

Com a crescente na quantidade de dados genômicos disponíveis, o alinhamento de sequências destaca-se como uma das tarefas relevantes no contexto da Bioinformática, cujos resultados são utilizados no auxílio às análises e posteriores inferências sobre esses dados. Assim, diversos algoritmos para alinhamento múltiplo de sequências baseados em diferentes heurísticas, como a de Algoritmos Genéticos, Otimização por Colônia de Formigas, Recozimento Simulado (*Simulated Annealling*), Busca Tabu, entre outros, foram propostos. No entanto, estudos apontam que com o uso de uma função objetivo mais adequada para aferir a qualidade do alinhamento produzido em cada caso específico, geralmente, produzem-se resultados com maior significância biológica. Além disso, o uso simultâneo de diferentes heurísticas para alinhamento múltiplo de sequências também tende a produzir melhores resultados, de modo que essa hibridização amenize as desvantagens de cada estratégia. No presente trabalho, implementou-se um escalonador automático de funções objetivo para selecionar o modelo de avaliação mais adequado para cada caso, com base na similaridade das sequências de entrada. Além disso, foi implementada uma fase de pós processamento com Otimização por Colônia de Formigas junto à MSA-GA, a fim de se refinar os alinhamentos produzidos pela ferramenta. Nos testes, pode-se verificar que o escalonador foi capaz de calcular de maneira adequada a similaridade dos conjuntos de entrada e, com isso, selecionar automaticamente a função objetivo mais apropriada. Além disso, com a execução da etapa de pós processamento, conseguiu-se refinar os alinhamentos produzidos pela MSA-GA, de modo a se obter resultados mais precisos.

Palavras-chave: Bioinformática. Alinhamento Múltiplo de Sequências. Algoritmos Genéticos.

Abstract

Due the increasing of the genomic data available, the multiple sequence alignment became one of the most important tasks in Bioinformatics, whose results are used to help biologists in their analysis. Thus, many algorithms to perform multiple sequence alignment based on different heuristics have been proposed, as Genetic Algorithms, Ant Colony, Simulated Annealing, Tabu Search, among others. Nonetheless, studies show that the use of an objective function suited for each case, generally, results in alignments with more biological significance. Moreover the simultaneous use of different heuristics to perform multiple sequence alignment can produce better results, smoothing the disadvantages of each strategy. In the present work, it was developed an automatic scheduler of objective functions to select the evaluation model more suited for each case, based on the similarity of the input sequences. Moreover it was implemented into the MSA-GA tool a post-processing stage with Ant Colony, in order to refine the obtained alignments. In the tests, it can be noticed that the scheduler calculates satisfactorily the similarity of the sequence sets and selects the more suited objective function. Moreover, with the post-processing stage, it was possible to refine the alignments of the MSA-GA, producing better results in terms of biological significance.

Keywords: Bioinformatics. Multiple Sequence Alignment. Genetic Algorithms.

Sumário

Sumário **x**

Lista de Figuras **xiii**

Lista de Tabelas **xv**

Lista de Abreviações **xvi**

1 Introdução **1**

1.1 Considerações Iniciais 1

1.2 Justificativa 2

1.3 Motivação e Escopo 4

1.4 Objetivos 5

1.5 Organização do Trabalho 5

2 Revisão Bibliográfica **7**

2.1 Considerações Iniciais 7

2.2 Biologia Molecular 7

2.2.1 Organização Celular 7

2.2.2 DNA, RNA e Proteínas 8

2.2.3 Filogenia e Padrões 10

2.3 Alinhamento de Sequências 11

2.3.1 Algoritmo de Needleman-Wunsch 12

2.3.2 Algoritmo de Smith-Waterman 14
2.4 Alinhamento Múltiplo de Sequências 16
2.4.1 Recozimento Simulado (*Simulated Annealing*) 17
2.4.2 Busca Tabu 18
2.4.3 Alinhamento Progressivo 19
2.4.4 Algoritmos Genéticos 21
2.4.5 Otimização por Colônia de Formigas 24
2.4.6 Heurísticas Híbridas 26
2.5 Ferramentas para Alinhamentos Múltiplos de Sequências 29
2.5.1 Clustal Omega 29
2.5.2 DIALIGN 30
2.5.3 MAFFT 32
2.5.4 SAGA 33
2.5.5 MSA-GA 34
2.6 Considerações Finais 39

3 Desenvolvimento do Trabalho 40
3.1 Considerações Iniciais 40
3.2 Escalonador Automático Multifunção 40
3.3 Pós Processamento com Otimização por Colônia de Formigas 46
3.3.1 Inicialização da Intensidade de Feromônio 47
3.3.2 Processamento do Deslocamento da Formiga 47
3.3.3 Cálculo da Pontuação do Alinhamento par-a-par e Atualização do Feromônio 50
3.3.4 Alinhamento Múltiplo de Sequências com o Algoritmo *Center Star* 52
3.4 Considerações Finais 53

4 Testes e Resultados **55**

4.1 Considerações Iniciais 55

4.2 Plataforma de Testes 55

4.3 Escalonador de Funções Objetivo 57

4.4 Refinamento do Alinhamento com Otimização por Colônia de Formigas 60

4.5 Considerações Finais 62

5 Conclusão **64**

5.1 Conclusões Gerais 64

5.2 Atividades Futuras 65

5.3 Publicações 66

Referências Bibliográficas **67**

Lista de Figuras

2.1 Interligação entre as bases que compõem o DNA. Fonte (ZAFALON, 2009). 9

2.2 Dogma central da biologia. Fonte: (VÊNCIO; OLIVEIRA, 2006). . . 9

2.3 Alinhamento global e alinhamento local de sequências de nucleotídeos. Fonte: (PROSDOCIMI et al., 2002). 13

2.4 Exemplo de alinhamento de sequências de nucleotídeos com o algoritmo de Needleman-Wunsch. Fonte: (ZAFALON, 2014 apud CRISTINO, 2012). 14

2.5 Construção da matriz de valores do algoritmo de Smith-Waterman. Fonte: (ZAFALON, 2014 apud CRISTINO, 2012). 15

2.6 Exemplo de árvore filogenética. Fonte: (ZAFALON, 2009). 20

2.7 Esquema da busca por comida em uma colônia de formigas. 25

2.8 Fluxograma da utilização da heurística de AG junto com a OCF. Adaptado de (LEE et al., 2008). 28

2.9 Cálculo de diagonal no DIALIGN. Fonte: (MORGENSTERN et al., 1998). 31

2.10 Esquema do algoritmo implementado na SAGA. Adaptado de: (NOTREDAME; HIGGINS, 1996). 34

2.11 Algoritmo para preenchimento da população inicial da MSA-GA. Fonte: (GONDRO; KINGHORN, 2007). 36

2.12 Esquema da função objetivo COFFEE. Adaptado de: (NOTREDAME; HOLM; HIGGINS, 1998). 38
2.13 Operadores de a) Recombinação Horizontal e b) Recombinação Vertical. Adaptado de: (GONDRO; KINGHORN, 2007). 39

3.1 Dados do conjunto de sequências *1idy* junto ao *BAliBase*. 41
3.2 Fluxograma do escalonador automático de funções objetivo. 43
3.3 Parâmetros da ferramenta MSA-GA. 44
3.4 Opções para definição manual de funções objetivo junto à MSA-GA. . 45
3.5 Exemplo de alinhamento com a estrutura de *grid* simplificado. 46
3.6 Escolha do *grid* superior. 48
3.7 Escolha do *grid* inferior. 48
3.8 Final efetivo na sequência inferior. 50
3.9 Exemplo de sequências a serem alinhadas com o *Center Star*. 53
3.10 Alinhamento múltiplo obtido com as sequências do exemplo. 53

4.1 Percentuais de identidade dos conjuntos de sequências menos similares. 58
4.2 Percentuais de identidade dos conjuntos de sequências mais similares. 59
4.3 Pontuações obtidas pela MSA-GA padrão e pela MSA-GA OCF. . . . 61

Lista de Tabelas

2.1 Os vinte aminoácidos que são codificados. 10

4.1 Valores dos parâmetros utilizados no Algoritmo Genético. 56

4.2 Valores dos parâmetros utilizados na Otimização por Colônia de Formigas. 56

4.3 Percentual de divergência de similaridades entre os conjuntos menos similares. 58

4.4 Percentual de divergência de similaridades entre os conjuntos mais similares. 59

4.5 Pontuações obtidas para cada caso de teste. 61

4.6 Desvios padrão das execuções de cada caso de teste. 62

Lista de Abreviações

AG Algoritmos Genéticos

AMS Alinhamento Múltiplo de Sequências

COFFEE *Consistency based Objective Function For alignmEnt Evaluation*

DNA *Deoxyribonucleic Acid*

FFT *Fast Fourier Transform*

GPU *Graphics Processing Unity*

HMM *Hidden Markov Model*

LOCQSPEC Localizador de *Quasispecies*

MAFFT *Multiple Alignment using Fast Fourier Transform*

NGS *Next Generation Sequencing*

OCF Otimização por Colônia de Formigas

RNA *Ribonucleic Acid*

SAGA *Sequence Alignment by Genetic Algorithm*

WSP *Weighted Sum-of-Pairs*

Capítulo 1

Introdução

1.1 Considerações Iniciais

Os avanços dos estudos na área da biologia, principalmente em genômica, propiciaram evoluções consideráveis tanto no aspecto científico quanto tecnológico, o que possibilitou o entendimento de diversos organismos, desde os micro-organismos, como os vírus e as bactérias, até àqueles mais complexos, como os seres humanos (EDGAR; BATZOGLOU, 2006). Muitas áreas se beneficiaram com esta evolução, em especial, a genômica, pois tornou-se possível a compreensão dos mecanismos de funcionamento do corpo humano (KHURI, 2008). Essa percepção apresenta-se como um fator de suma importância em diversos cenários, principalmente no auxílio à eventual cura de diferentes tipos de doenças (KHURI, 2008).

A partir da descoberta da estrutura básica do DNA (WATSON; CRICK et al., 1953) e com os diversos avanços tecnológicos, a quantidade de dados biológicos disponíveis passou a crescer consideravelmente, de modo que se tornou infactível a análise manual dessa quantidade de informações (ZAFALON et al., 2015). Com isso, fez-se necessária a busca por estratégias computacionais que auxiliassem os trabalhos de análise por parte dos biólogos, para que dessa forma posteriores inferências possam ser feitas de maneira satisfatória. Estes fatores foram pontos importantes que originaram, por

conseguinte, a Bioinformática (KHURI, 2008).

Assim, a Bioinformática destaca-se como uma área da ciência da computação que visa oferecer soluções computacionais para problemas pertinentes ao escopo da biologia (EDGAR; BATZOGLOU, 2006). A importância da Bioinformática se deve, principalmente, ao fato dos biólogos necessitarem do poder computacional para a análise satisfatória dos dados que, geralmente, são obtidos em sua forma bruta e que precisam ser processados para a realização de posteriores inferências (KHURI, 2008).

1.2 Justificativa

Existem inúmeras ferramentas de Bioinformática desenvolvidas para auxiliar os biólogos em diferentes frentes. No entanto, dentre as principais áreas de atuação da Bioinformática, destacam-se as ferramentas para alinhamento de sequências (EDGAR; BATZOGLOU, 2006) e reconhecimento de padrões (KHURI, 2008), cujos resultados produzidos apoiam a detecção de importantes regiões, também conhecidas como *hotspots* (pontos-quentes). Estes pontos geralmente são relevantes pois podem conter informações sobre determinados genótipos e, consequentemente, auxiliar na descoberta de possíveis curas para diferentes tipos de doenças (LIEW; YAN; YANG, 2005; ZHOU et al., 2007).

Muitos esforços foram voltados ao desenvolvimento de diversas técnicas para se executar alinhamentos de sequências com resultados satisfatórios. Sabe-se que o melhor alinhamento pode ser obtido de maneira determinística, por meio de algoritmos de programação dinâmica, como Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970) e Smith-Waterman (SMITH; WATERMAN, 1981). No entanto, estes algoritmos foram idealmente desenvolvidos para executarem alinhamentos de pares de sequências e, devido à complexidade computacional de ordem quadrática destas técnicas, torna-se infactível alinhar conjuntos que contenham mais do que duas sequências (WANG; JIANG, 1994).

Com isso, devido à grande quantidade de dados atualmente disponíveis, fez-se necessário o desenvolvimento de técnicas para alinhamentos múltiplos de sequências (AMS) (MORRISON; MORGAN; KELCHNER, 2015). Estas técnicas podem ser baseadas em diversas heurísticas (EDGAR; BATZOGLOU, 2006), em que uma das mais utilizadas atualmente é a de alinhamento progressivo, implementada em ferramentas muito conhecidas na área da Bioinformática, como as da família Clustal (THOMPSON; HIGGINS; GIBSON, 1994; SIEVERS et al., 2011). Além da heurística de alinhamento progressivo, diversas abordagens estocásticas têm sido aplicadas com sucesso para a resolução de problemas de alinhamentos múltiplos de sequências, como as baseadas em Recozimento Simulado (*Simulated Annealig*) (SCHWARTZ; PACHTER, 2007), Busca tabu (RIAZ; WANG; LI, 2004), Otimização por Colônia de Formigas (OCF) (MOSS; JOHNSON, 2003), Algoritmos Genéticos (AG) (NOTREDAME; HIGGINS, 1996; GONDRO; KINGHORN, 2007), entre outras.

No entanto, as heurísticas anteriormente citadas geralmente apresentam pontos favoráveis e desfavoráveis e, assim, o desenvolvimento de estratégias que sejam capazes de conciliar bom desempenho com resultados de significância biológica relevantes ainda apresenta-se como um desafio nos dias atuais (MORRISON; MORGAN; KELCHNER, 2015).

Dessa forma, uma tendência notável na área de Bioinformática é a técnica de hibridização de heurísticas para alinhamentos múltiplos de sequências (LIU; POPP; SCHMIDT, 2014; GHOUZAM et al., 2015; KELLER et al., 2011; PIERRI; PARISI; PORCELLI, 2010; SUN et al., 2012). Com isso, observa-se que a escolha da melhor estratégia para casos específicos, aliada à hibridização de heurísticas, a fim de se amenizar os pontos desfavoráveis da abordagem utilizada, tende a produzir resultados com maior significância biológica (LEE et al., 2008).

1.3 Motivação e Escopo

Nos dias atuais, a busca pela melhor ferramenta para alinhamento múltiplo de sequências continua sendo foco de muitas pesquisas na área de Bioinformática, principalmente no que diz respeito ao uso de heurísticas mais adequadas para cada problema. Dentre as diversas heurísticas existentes, o Algoritmo Genético destaca-se, pois enquadra-se bem aos problemas dos alinhamentos múltiplos.

Existem diversas ferramentas para alinhamento múltiplo de sequências baseadas em AG, como a SAGA (*Sequence Alignment by Genetic Algorithm*) (NOTREDAME; HIGGINS, 1996), a MSA-GA (GONDRO; KINGHORN, 2007), ferramentas com funções multiobjetivo (KAYA; KAYA; ALHAJJ, 2016), entre outras. Nesse contexto, a MSA-GA destaca-se pois utiliza uma abordagem mais simples de AG e, ainda assim, é capaz de produzir melhores resultados quando comparada a outras ferramentas muito utilizadas em Bioinformática, como a ClustalW (GONDRO; KINGHORN, 2007).

No entanto, a função objetivo padrão que afere a qualidade dos alinhamentos produzidos pela MSA-GA é a *Weighted Sum-of-Pairs* (WSP), a qual tende a produzir resultados de menor qualidade para conjuntos de sequências menos similares (AMORIM et al., 2015). Para amenizar esse problema, Amorim et al. (2015) implementaram a função COFFEE (*Consistency based Objective Function for alignmEnt Evaluation*) junto à MSA-GA, o que permitiu que a ferramenta produzisse melhores resultados para conjuntos menos correlatos. Porém, Amorim et al. (2015) observaram em seus resultados que, embora a função COFFEE seja mais adequada para avaliar conjuntos de sequências menos similares, a função WSP ainda tende a produzir melhores resultados para conjuntos mais correlatos.

Assim, a implementação de um escalonador automático de funções objetivo junto à MSA-GA, com base na similaridade do conjunto de sequências de entrada, pode ajudar a ferramenta a selecionar, durante sua execução, a função objetivo mais adequada para cada caso.

Além disso, sabe-se que devido à natureza do AG, os alinhamentos produzidos podem se direcionar a um ponto de máximo local, o que significa que, nesses casos, ainda se pode obter um alinhamento de maior significância biológica (LEE et al., 2008). Assim, o uso de uma outra heurística como etapa de pós processamento pode refinar os resultados produzidos pelo AG.

1.4 Objetivos

Embora as modificações realizadas por Amorim et. al. (2015) tenham proporcionado melhorias aos alinhamentos de conjuntos de sequências pouco similares produzidos pela MSA-GA, percebe-se que a função WSP ainda rende melhores resultados para conjuntos de sequências mais similares.

Assim, o presente trabalho tem como objetivo a implementação de um escalonador automático, baseado na similaridade das sequências de entrada. Este escalonador irá selecionar uma das funções objetivo previamente implementadas na MSA-GA (WSP ou COFFEE), de acordo com a similaridade calculada por um modelo matemático.

Além disso, o presente trabalho objetiva a implementação de uma etapa de pós processamento junto à MSA-GA, com base em OCF. Os alinhamentos produzidos pelo AG da ferramenta serão dados como entrada para a OCF, de modo a refinar os resultados que eventualmente encontram-se em um ponto de máximo local.

1.5 Organização do Trabalho

O presente trabalho está dividido em cinco capítulos, incluindo o atual. No Capítulo 2 é realizada uma revisão bibliográfica referente aos assuntos pertinentes ao escopo do trabalho de dissertação que será desenvolvido, elucidando o estado da arte das técnicas. No Capítulo 3 apresenta-se o desenvolvimento do trabalho no que diz respeito ao módulo do escalonador de funções objetivo e também à etapa de pós processa-

mento com OCF. No Capítulo 4 são exibidos os testes e resultados obtidos. Por fim, no Capítulo 5, apresentam-se as conclusões e as atividades futuras relacionadas com o presente trabalho.

Capítulo 2

Revisão Bibliográfica

2.1 Considerações Iniciais

Este capítulo tem o objetivo de apresentar conceitos fundamentais na área de Bioinformática, desde fundamentos da biologia molecular, até estratégias para alinhamentos de sequências. Estes conceitos são importantes para a compreensão do estado da arte que envolve a Bioinformática.

2.2 Biologia Molecular

Para que haja o entendimento do escopo que envolve a área de Bioinformática, é necessária a apresentação de alguns conceitos elementares em Biologia Molecular, tais como organização molecular, macromoléculas biológicas, filogenia e padrões.

2.2.1 Organização Celular

Ao longo do tempo, esforços foram voltados aos estudos na área da biologia em que se buscou o entendimento da grande variedade dos organismos existentes. Como resultado destes estudos, observou-se que, independente do tipo de organismo considerado,

quando se trata de seres vivos, a célula é a unidade que representa as estruturas e funções do indivíduo (LEMOS; ARAGAO; CASANOVA, 2003).

Sabe-se que as células são divididas entre os grupos dos eucariontes e dos procariontes (RUBIN et al., 2000; WHITMAN; COLEMAN; WIEBE, 1998), em que a principal diferença entre estes é que o dos eucariontes é composto por células que apresentam envoltório nuclear, enquanto as células procariontes não partilham desta característica.

No entanto, o estudo celular tem evoluído de maneira considerável, principalmente na área da genômica, desde a descoberta da estrutura básica do DNA (*Deoxyribonucleic Acid*) . Atualmente, o DNA apresenta-se, juntamente com o RNA (*Ribonucleic Acid*) e as proteínas, como uma macromolécula de suma importância para o entendimento do funcionamento de diversos organismos. Isto torna necessário um estudo mais detalhado para a compreensão das características de cada indivíduo (ADAMS et al., 1992; BAJORATH; STENKAMP; ARUFFO, 1993).

2.2.2 DNA, RNA e Proteínas

O entendimento das macromoléculas de DNA, RNA e também das proteínas é imprescindível para a compreensão das estruturas e funções de cada organismo. No escopo da Bioinformática, estas macromoléculas são chamadas de biossequências ou sequências biológicas (EIDHAMMER; JONASSEN; TAYLOR, 2000; LEMOS; ARAGAO; CASANOVA, 2003).

Assim, tanto as sequências de DNA como as de RNA são classificadas como sequências de nucleotídeos. Segundo estudos, a estrutura básica do DNA, que se encontra representada na Figura 2.1, é composta por uma fita dupla de nucleotídeos e quatro bases nitrogenadas, as quais podem ser Adenina (A), Timina (T), Citosina (C) e Guanina (G), que se ligam por meio de uma relação conhecida como ponte de hidrogênio (WATSON; CRICK et al., 1953).

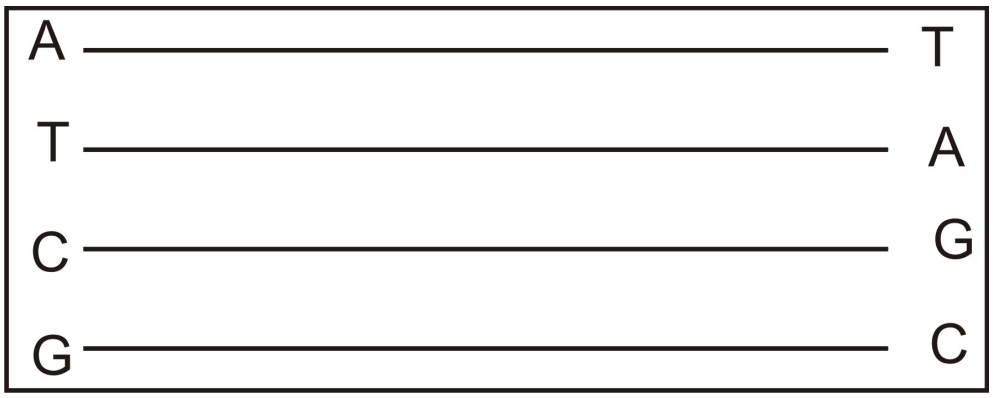


Figura 2.1: Interligação entre as bases que compõem o DNA. Fonte (ZAFALON, 2009).

O RNA, por sua vez, trata-se de uma fita simples e, assim como o DNA, também é formado por quatro bases nitrogenadas. No entanto, quando há a transcrição do RNA, substituem-se as posições de Timina (T) por Uracila (U).

Todas as informações dos indivíduos são armazenadas em seu DNA (LEMOS; BASÍLIO; CASANOVA, 2003). Estas informações são então transcritas para as moléculas de RNA, as quais contém o código para a ordenação dos aminoácidos. A partir disso, é possível sintetizar diferentes proteínas por meio de um processo que envolve a tradução do RNA (ZAFALON, 2009), conforme observa-se na Figura 2.2.

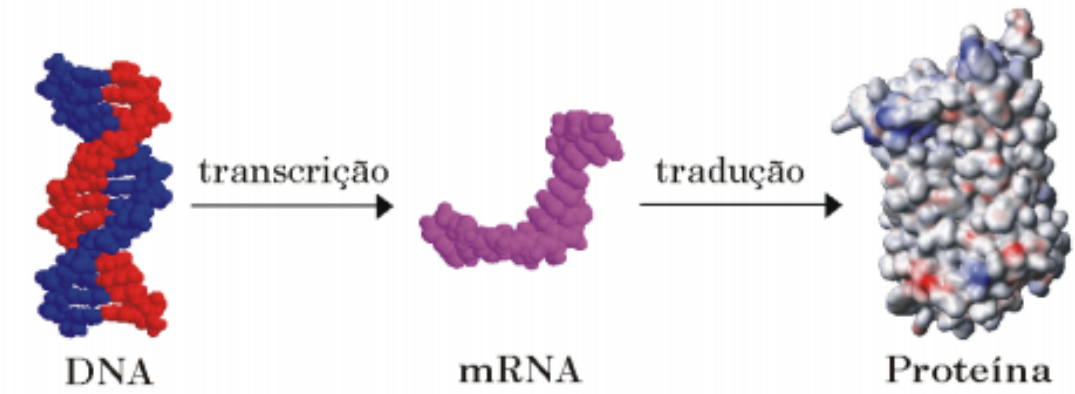


Figura 2.2: Dogma central da biologia. Fonte: (VÊNCIO; OLIVEIRA, 2006).

Desta forma, os aminoácidos, que são obtidos a partir de um códon (sequência de três bases nitrogenadas) resultante do processo da síntese de RNA, compõem um grupo de vinte itens essenciais que podem ser codificados para a formação de proteínas,

conforme apresenta-se na Tabela 2.1 (LEMOS; CASANOVA, 2000).

Tabela 2.1: Os vinte aminoácidos que são codificados.

Nome	Símbolo	Abreviação
Glicina ou Glicocola	Gly, Gli	G
Alanina	Ala	A
Leucina	Leu	L
Valina	Val	V
Isoleucina	Ile	I
Prolina	Pro	P
Fenilalanina	Phe ou Fen	F
Serina	Ser	S
Treonina	Thr, The	T
Cisteina	Cys, Cis	C
Tirosina	Tyr, Tir	Y
Asparagina	Asn	N
Glutamina	Gln	Q
Aspartato ou Ácido aspártico	Asp	D
Glutamato ou Ácido glutâmico	Glu	E
Arginina	Arg	R
Lisina	Lys, Lis	K
Histidina	His	H
Triptofano	Trp, Tri	W
Metionina	Met	M

2.2.3 Filogenia e Padrões

Com os avanços dos estudos e tecnologias nas áreas da biologia molecular e celular, possibilitou-se o entendimento de uma série de fatores e características que são propagadas pelos diversos seres vivos ao longo das gerações.

Neste âmbito, a filogenia e a análise de padrões são áreas que se preocupam em identificar, respectivamente, o caminho de determinado processo evolutivo e quais as características que foram predominantemente preservadas pelos organismos ao longo das gerações (GOULD, 1977). Além disso, a descoberta de padrões pode ser útil para a compreensão de estruturas e para a classificação de famílias de proteínas desconhecidas (ZAFALON, 2009).

Atualmente, existem diversas ferramentas e algoritmos para reconhecimento de padrões em biossequências (LIEW; YAN; YANG, 2005). Diferentes heurísticas, tais como Modelo de Markov têm sido aplicadas com êxito em Bioinformática (FINK, 2014). Marucci (MARUCCI et al., 2008), por exemplo, desenvolveu a ferramenta LOCQSPEC (Localizador de *Quasispecies*) com o intuito de realizar o reconhecimento de padrões em determinadas *quasispecies* de modo que outras ferramentas não eram capaz de processar de maneira satisfatória. Todos esses esforços desprendidos na vertente de reconhecimento de padrões corroboram a importância da técnica para a área de Bioinformática.

No entanto, devido ao crescente volume de dados genômicos disponíveis nos últimos anos, a análise de filogenia e padrões dessa grande quantidade de informações tornou-se computacionalmente custosa. Com isso, o uso de ferramentas e algoritmos para análises biológicas, tais como estratégias de alinhamento de sequências, tornou-se imprescindível para a realização de posteriores inferências sobre esses dados (EDGAR; BATZOGLOU, 2006).

2.3 Alinhamento de Sequências

As técnicas de alinhamento de sequências não são apenas algoritmos que auxiliam análises biológicas triviais. Essas estratégias possibilitam que biólogos realizem diferentes aferições nos mais variados grupos de biossequências, tanto no âmbito de quantidade de sequências quanto no quesito variedade biológica, de modo que a análise manual desses dados seria dispendiosa e infactível (ZAFALON, 2014).

Desta forma, o alinhamento de sequências apresenta-se como uma das tarefas de notável destaque na área de Bioinformática, visto que fornece ferramental computacional para análises de sequências biológicas. Esta tarefa é vastamente utilizada para reconstrução de árvores filogenéticas, predição de funções e estruturas de proteínas desconhecidas, além de auxiliar no processo de reconhecimento de padrões em

sequências biológicas (EDGAR; BATZOGLOU, 2006; KHURI, 2008).

Basicamente, as estratégias de alinhamento de sequências promovem rearranjos nas bases de nucleotídeos ou aminoácidos das biossequências, também conhecidas como resíduos. Além disso, possibilitam a inserção de espaços (*gaps*) com o intuito de se buscar uma configuração que reflita o maior nível de similaridade entre as sequências de entrada, segundo alguma função de medida de identidade (EDGAR; BATZOGLOU, 2006). Cada coincidência entre resíduos é classificada como *match* e cada diferença entre resíduos de sequências como *mismatch*. Durante o processo de alinhamento, a função que reflete a similaridade entre as sequências pontua, de maneira ponderada, a quantidade de *matches* e *mismatches*, de modo que ao final da execução, o algoritmo produza uma solução com a maior pontuação possível (EDGAR; BATZOGLOU, 2006; KHURI, 2008).

A melhor pontuação para alinhamentos de sequências pode ser obtida por meio de algoritmos de programação dinâmica, como os de Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970) e Smith-Waterman (SMITH; WATERMAN, 1981), em que o primeiro é utilizado em estratégias de alinhamento global, enquanto o segundo é utilizado para alinhamento local, conforme observa-se na Figura 2.3, a partir de um exemplo de alinhamento de sequências de nucleotídeos.

2.3.1 Algoritmo de Needleman-Wunsch

O algoritmo de Needleman-Wunsch é uma técnica utilizada para a execução de alinhamentos globais de sequências de nucleotídeos ou aminoácidos (NEEDLEMAN; WUNSCH, 1970). Isto significa que o algoritmo irá buscar por uma configuração com o maior nível de similaridade, baseando-se nas sequências como um todo. Assim, nos alinhamentos globais, não existe a preocupação com pontos em particular das sequências envolvidas (ZAFALON, 2014).

Como o algoritmo de Needleman-Wunsch utiliza uma abordagem determinística

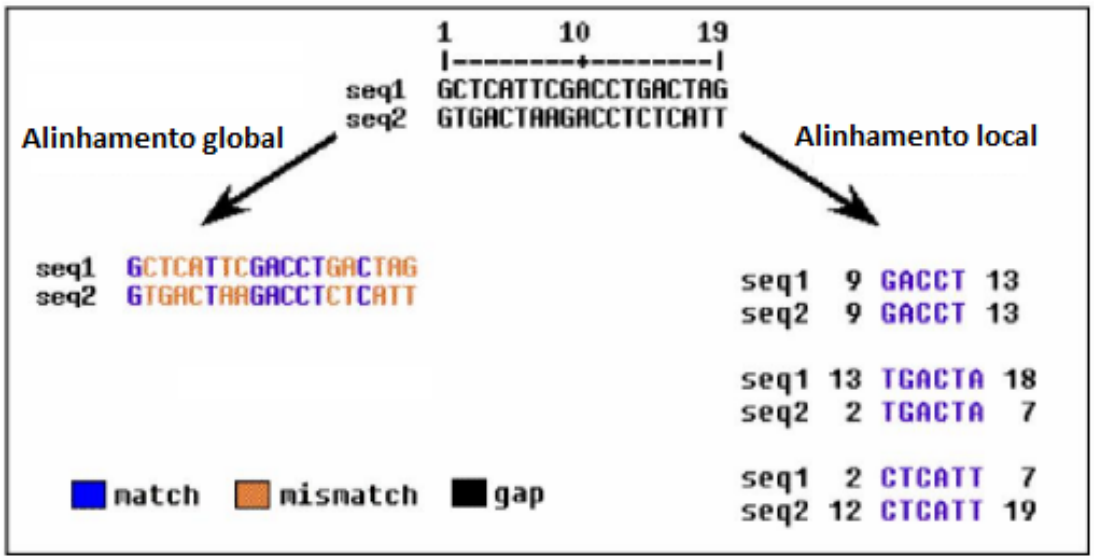


Figura 2.3: Alinhamento global e alinhamento local de sequências de nucleotídeos. Fonte: (PROSDOCIMI et al., 2002).

de programação dinâmica, é certo que o resultado obtido será sempre o melhor alinhamento possível para as sequências de entrada escolhidas (GUSFIELD, 1997).

Basicamente, o referido algoritmo realiza alinhamentos de sequências par-a-par, utilizando-se de uma matriz de pontuação, cujas células são preenchidas segundo funções matemáticas que assinalam valores positivos para posições de *match* e valores negativos para posições de *gap* e *mismatch*, o que permite que ao fim do processo se obtenha um resultado com as características desejadas (ZAFALON, 2014). Ao término do preenchimento da matriz, realiza-se uma escalada reversa (*backtracking*) para a obtenção do melhor alinhamento, conforme observa-se no exemplo da Figura 2.4 (ZAFALON, 2014).

O algoritmo de Needleman-Wunsch é uma estratégia para a busca de sequências similares como um todo. No entanto, existem situações em que, mesmo em pares de sequências menos correlatas, há a necessidade de busca por regiões específicas entre os pares de sequências considerados. Nestes casos, a utilização de algoritmos de alinhamentos locais é mais adequada (SMITH; WATERMAN, 1981).

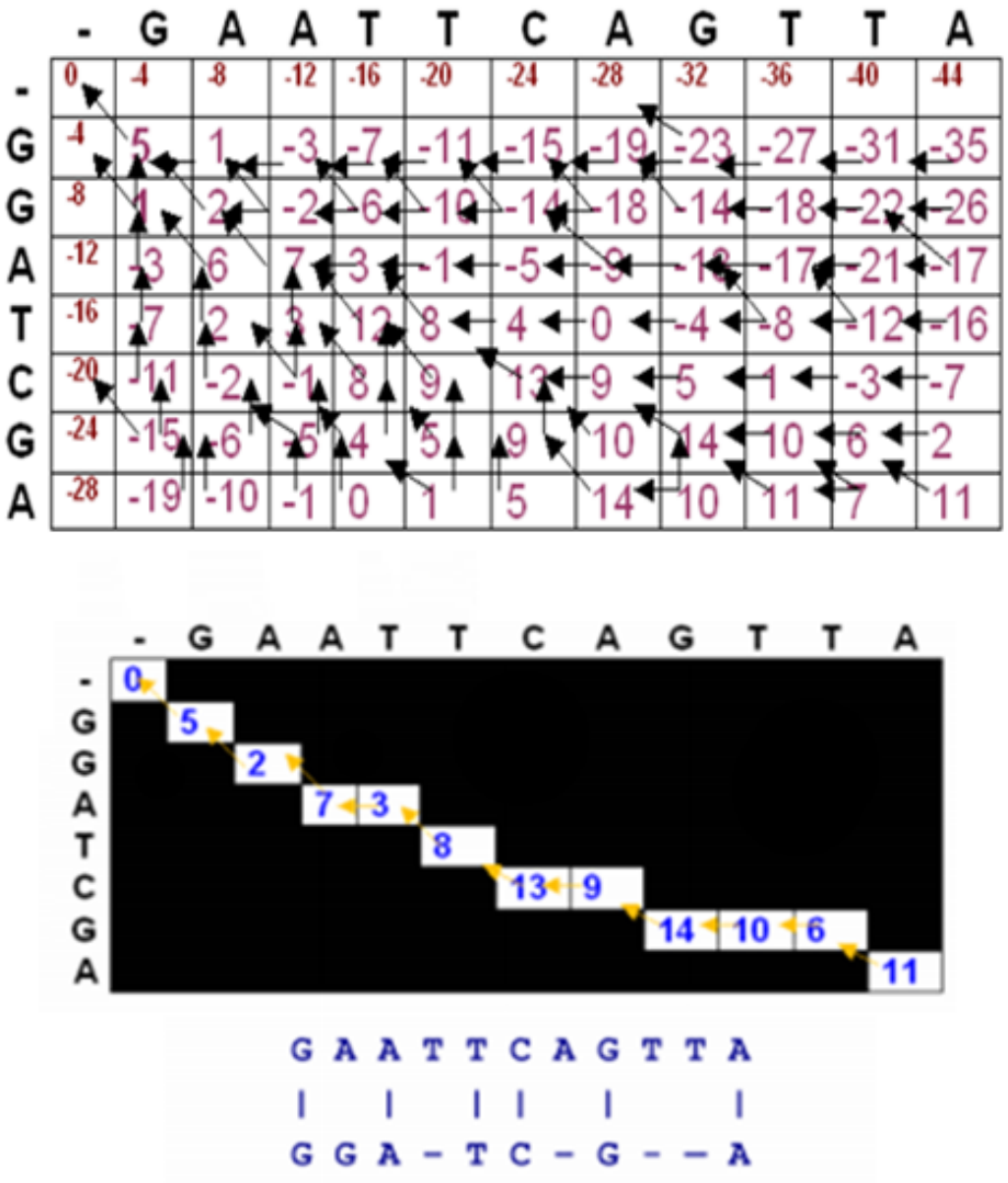


Figura 2.4: Exemplo de alinhamento de sequências de nucleotídeos com o algoritmo de Needleman-Wunsch. Fonte: (ZAFALON, 2014 apud CRISTINO, 2012).

2.3.2 Algoritmo de Smith-Waterman

Ao contrário dos algoritmos de alinhamentos globais, as estratégias de alinhamentos locais preocupam-se com pontos de similaridade específicos entre duas sequências analisadas. Estes pontos podem representar uma característica procurada por pesquisadores durante determinada análise (ZAFALON, 2014).

A estratégia mais difundida na área de Bioinformática para alinhamentos locais de

sequências é o algoritmo de Smith-Waterman (SMITH; WATERMAN, 1981). Este algoritmo baseia-se no de Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970) e, portanto, apresenta diversas características em comum.

Assim como na estratégia de Needleman-Wunsch, o algoritmo de Smith-Waterman utiliza-se de uma matriz de valores, a qual é preenchida segundo equações matemáticas que pontuam os *matches* e penalizam os *gaps* e *mismatches*. No entanto, a principal diferença entre os dois algoritmos é que o de Smith-Waterman não assume valores negativos na matriz de pontuação, sendo que para estes casos adota-se o valor 0, conforme demonstra-se na Figura 2.5 (ZAFALON, 2014).

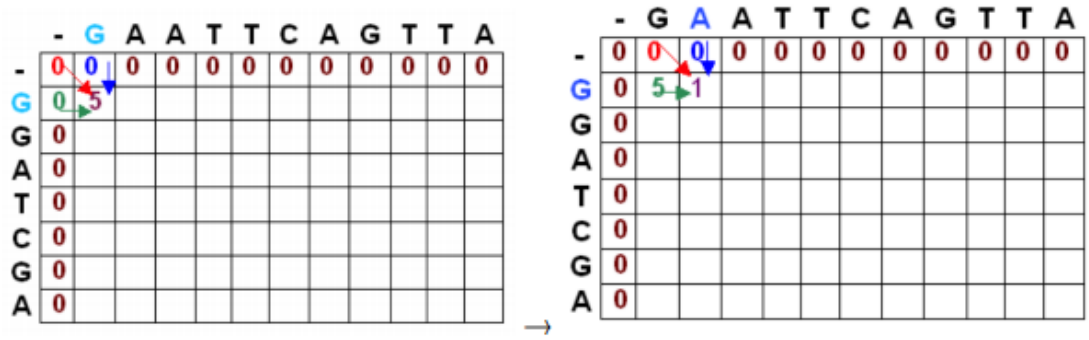


Figura 2.5: Construção da matriz de valores do algoritmo de Smith-Waterman. Fonte: (ZAFALON, 2014 apud CRISTINO, 2012).

Após o completo preenchimento da matriz de valores, o algoritmo promove a escalada reversa para a obtenção do alinhamento. A diferença em relação ao de Needleman-Wunsch é que a escalada se inicia no maior valor presente na matriz e finaliza ao encontrar qualquer valor zero. É importante ressaltar que, assim como o algoritmo de Needleman-Wunsch, a estratégia de Smith-Waterman utiliza algoritmo de programação dinâmica e, de maneira determinística, sempre obtém o melhor alinhamento para dadas sequências de entrada (ZAFALON, 2014).

Porém, os algoritmos de Needleman-Wunsch e Smith-Waterman foram idealmente desenvolvidos para alinhamentos de pares de sequências (AMORIM et al., 2015)), o que geralmente impossibilita a execução de alinhamentos que contenham três ou mais sequências (WANG; JIANG, 1994). Este fato apresenta-se como um grande problema

na conjuntura atual da Bioinformática, visto que cada vez mais deseja-se realizar alinhamentos de um grande número de sequências, dos mais variados organismos, simultaneamente.

Assim, fez-se necessário o desenvolvimento de estratégias de alinhamentos múltiplos de sequências, os quais baseiam-se em diversas heurísticas (EDGAR; BATZOGLOU, 2006). No entanto, a implementação de algoritmos que conciliem resultados com significância biológica e tempo de execução hábil continua sendo buscada pelos pesquisadores.

2.4 Alinhamento Múltiplo de Sequências

Embora os algoritmos de programação dinâmica obtenham sempre o melhor alinhamento para pares de sequências, a execução de alinhamentos com mais do que duas sequências é impraticável, devido à complexidade quadrática desta classe de algoritmos (WANG; JIANG, 1994).

Desta forma, diversos pesquisadores voltaram esforços ao desenvolvimento de algoritmos heurísticos. Estes possibilitaram a produção de alinhamentos de múltiplas sequências, geralmente com boa significância biológica e em um tempo de execução viável (EDGAR; BATZOGLOU, 2006).

Atualmente, existem inúmeras heurísticas para alinhamentos múltiplos de sequências que se destacam no contexto da Bioinformática, tais como a de Recozimento Simulado (*Simulated Annealing*) (SCHWARTZ; PACHTER, 2007), a Busca Tabu (*Tabu Search*) (RIAZ; WANG; LI, 2004), a de Alinhamento Progressivo (THOMPSON; HIGGINS; GIBSON, 1994; SIEVERS et al., 2011), a de Algoritmos Genéticos (NOTREDAME; HIGGINS, 1996; GONDRO; KINGHORN, 2007) e, finalmente, a Otimização por Colônia de Formigas (MOSS; JOHNSON, 2003).

2.4.1 Recozimento Simulado (*Simulated Annealing*)

A heurística de Recozimento Simulado é uma estratégia comumente utilizada em problemas de otimização que envolvem a busca por um máximo global (KIRKPATRICK et al, 1983).

Para a implementação deste algoritmo é necessário que se defina um espaço de estados $X = \{x_1, ..., x_n\}$ além de uma função de custo $C : X \Rightarrow R$, em que R é um conjunto de números reais a ser definido (ZAFALON, 2014). A ideia principal é encontrar um estado X_{opt} ao longo das iterações do algoritmo, as quais baseiam-se em regras definidas através de um fator de decaimento da temperatura T, que inicialmente assume um alto valor e tende a diminuir segundo o escalonamento do algoritmo. Este fator é responsável pelo controle da probabilidade de aceitação de um estado X_{novo}.

As possíveis regras de transição para aprovação de um novo estado na heurística de Recozimento Simulado são (KIRKPATRICK et al., 1983):

1. Se $\Delta E \leq 0$, aceita um novo estado X_{novo} com ΔE sendo a variação de energia.

2. Se $\Delta E > 0$, aceita um novo estado X_{novo}, com probabilidade $P(\Delta E) = e^{-\Delta E/T}$ em que T é a temperatura e $\Delta E = C(x_{novo}) - C(x_{atual})$ é a diferença de custo.

Neste caso, a probabilidade $P(\Delta E)$ é importante pois impede que o algoritmo caia num estado de mínimo local, o que significa que não será possível gerar algum estado X_{novo} que tenha um custo menor do que o X_{atual}.

Em Bioinformática, as primeiras aplicações de Recozimento Simulado foram em técnicas de alinhamento múltiplo de sequências (KIM; PRAMANIK; CHUNG, 1994), em que se utilizou de estratégias de paralelismo para se amenizar o problema do alto custo computacional (ISHIKAWA et al., 1993). Os bons resultados obtidos ampliaram a gama de aplicações de SA para a obtenção de estruturas terciárias de proteínas (SIMONS et al., 1997).

Embora seja uma técnica que tem sido trabalhada há algum tempo no contexto da Bioinformática, atualmente ainda é possível observar alguns esforços voltados para a melhoria da heurística de Recozimento Simulado, como a paralelização de processos utilizando grids computacionais (ZHU et al., 2011; MAREUIL et al., 2011). Além destes, Correa adaptou a técnica de SA a uma abordagem que utiliza fatores de pesos para classificar os resíduos das sequências e obter melhores alinhamentos (CORREA et al., 2012).

Apesar da vasta utilização no meio científico, a heurística de Recozimento Simulado esbarra na desvantagem do alto custo computacional, devido ao fato de se basear no método de Monte Carlo (KALOS, 2007). No entanto, vale a pena ressaltar que a utilização de alguma outra estratégia aliada ao SA tende a amenizar este problema (ZAFALON, 2014). Além disso os ótimos resultados obtidos pelo algoritmo geralmente compensam os esforços computacionais empregados na sua execução.

2.4.2 Busca Tabu

A Busca Tabu é uma heurística de otimização que se baseia em características de otimização linear e que utiliza estratégias de inteligência artificial. A aplicação desta heurística é indicada para a solução de problemas de decisões críticas que necessitam de uma vasta busca no seu espaço de soluções (GLOVER; LAGUNA, 1999).

Em Bioinformática, nota-se a aplicação de Busca Tabu em diversos contextos, como por exemplo para alinhamento global de biossequências (RIAZ; WANG; LI, 2004) ou mesmo numa fase de pós-processamento, para a tentativa de refinamento em descoberta de padrões (RIAZ; YI; LI, 2005).

Atualmente, a Busca Tabu tem sido vastamente utilizada em problemas de Bioinformática, em especial naqueles envolvidos com estruturas de proteínas. Lin e Zhang realizaram um ajuste no algoritmo de Busca Tabu e o aplicaram para a predição de estruturas proteicas (LIN; ZHANG et al., 2014). De maneira similar, Cavuslar utilizou

Busca Tabu como abordagem para a solução de problemas de atribuição baseados em estruturas de proteínas (CAVUSLAR; CATAY; APAYDIN, 2012).

2.4.3 Alinhamento Progressivo

Devido ao alto custo computacional de se realizar alinhamentos múltiplos de sequências a partir de algoritmos de programação dinâmica, diversas heurísticas baseadas em fatores probabilísticos foram desenvolvidas (AMORIM et al., 2015).

Dentre as estratégias disponíveis para alinhamento múltiplo de sequências, uma das mais utilizadas em Bioinformática é a heurística de Alinhamento Progressivo (FENG; DOOLITTLE, 1987), visto que é capaz de produzir resultados com boa significância biológica em um tempo viável. Esta técnica foi utilizada em ferramentas amplamente conhecidas no contexto da Bioinformática, como as da família Clustal (THOMPSON; HIGGINS; GIBSON, 1994; SIEVERS; HIGGINS, 2014) e a T-COFFEE (NOTREDAME; HIGGINS; HERINGA, 2000).

A heurística de Alinhamento Progressivo pode ser dividida em três fases bem definidas (WALLACE; BLACKSHIELDS; HIGGINS, 2005; ZAFALON, 2014):

1. **Alinhamento par-a-par e construção da matriz de pontuação:** nesta etapa, alinha-se cada par de sequências do conjunto de entrada e calcula-se as distâncias destes pares com algoritmos de programação dinâmica. Vale ressaltar que para um conjunto com n sequências, haverá $n*(n-1)/2$ pares. Posteriormente, declara-se uma matriz de pontuação, a qual é preenchida com os valores das distâncias previamente calculados;

2. **Construção da árvore guia:** após o cálculo das distâncias, declara-se uma estrutura de dados do tipo árvore, a qual representa a árvore filogenética do conjunto de sequências em questão e que será utilizada na etapa de alinhamento. Existem diversos métodos para a construção da árvore, como por exemplo, o de

Neighbor-Joining (SAITOU; NEI, 1987). Este método junta sucessivamente as sequências mais similares, conforme observa-se na Figura 2.6.

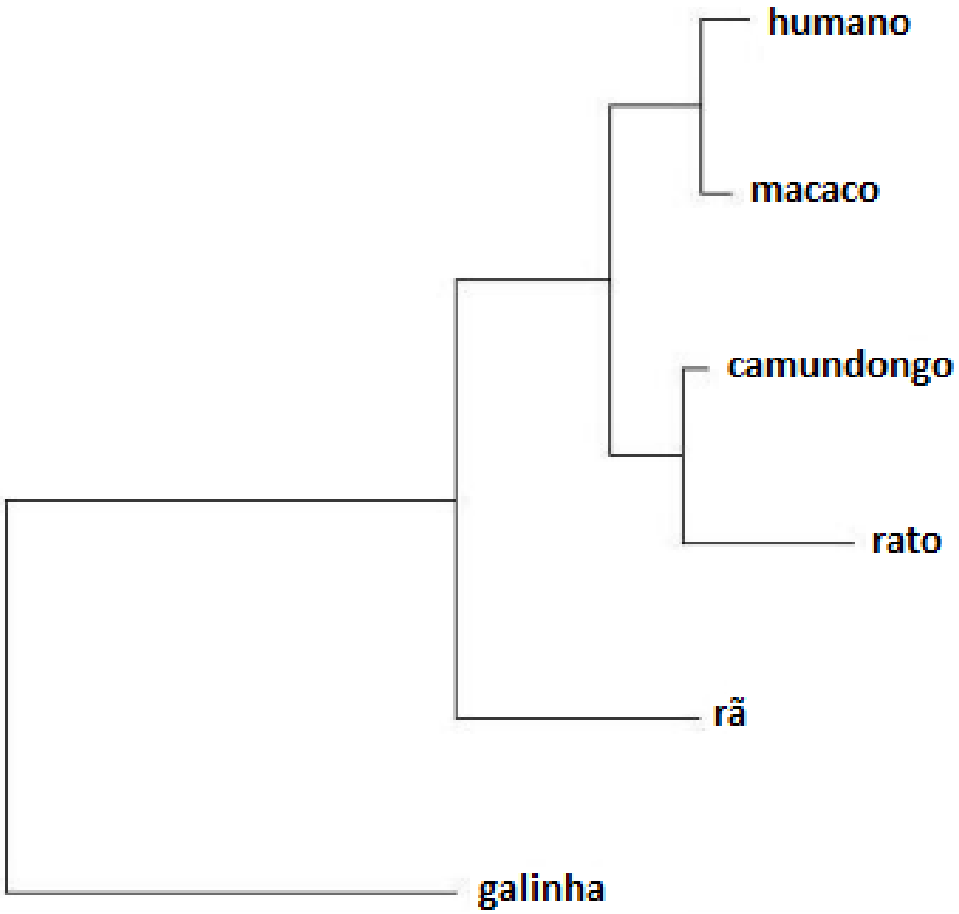


Figura 2.6: Exemplo de árvore filogenética. Fonte: (ZAFALON, 2009).

Nesse caso, o humano e o macaco são os organismos mais similares e, portanto, são posicionados mais próximos junto à árvore;

3. **Alinhamento múltiplo:** uma vez que as sequências foram alocadas na árvore filogenética, o algoritmo Progressivo iniciará o alinhamento pelas duas sequências mais similares da árvore. A partir deste alinhamento obtido, incorpora-se a próxima sequência mais similar, segundo a árvore filogenética, e obtém-se um novo alinhamento. Este processo se repete por $n-1$ vezes, em que n é o número de sequências a serem alinhadas, até que o alinhamento final seja obtido.

Embora o Alinhamento Progressivo seja capaz de produzir resultados com boa

significância biológica em um tempo factível, este algoritmo, geralmente, não produz bons resultados quando processa um número considerável de sequências devido à natureza da estratégia. Acrescenta-se a isso o fato de ter uma dependência da ordem em que as sequências estão posicionadas no conjunto de entrada, o que afeta a etapa de construção da árvore guia e, consequentemente, a qualidade dos resultados obtidos (BOYCE; SIEVERS; HIGGINS, 2015).

No entanto, diversas pesquisas encontram-se voltadas para a melhoria da heurística de Alinhamento Progressivo, conforme observa-se na ferramenta Clustal Omega (SIEVERS; HIGGINS, 2014), que permite a obtenção de bons alinhamentos mesmo com um número considerável de sequências. Além disso, Boyce elucidou que o uso de árvores guia encadeadas pode amenizar os problemas da natureza do algoritmo (BOYCE; SIEVERS; HIGGINS, 2015).

2.4.4 Algoritmos Genéticos

Outra alternativa amplamente utilizada em problemas de busca e otimização é a heurística de Algoritmos Genéticos. Trata-se de uma técnica baseada na Teoria da Evolução (SCHWEFEL; RUDOLPH, 1995), em que os organismos que representam as possíveis soluções são expostos a processos de mutação, recombinação e seleção, a fim de evoluir a população de soluções candidatas, as quais são mensuradas por uma função objetivo (GOLDBERG; HOLLAND, 1988). Vale ressaltar que Algoritmos Genéticos são indicados para problemas que possuem um vasto espaço de soluções, em que se deseja obter uma solução ótima (GOLDBERG; HOLLAND, 1988).

Em Bioinformática, a heurística de AG é comumente utilizada para a solução de problemas de alinhamentos múltiplos de sequências, visto que geralmente permite a obtenção de bons resultados em um tempo factível (AMORIM et al., 2015).

Do ponto de vista computacional, cada indivíduo representa um possível alinhamento, em que a qualidade da solução é mensurada segundo alguma função objetivo

que reflita o nível de similaridade entre as sequências consideradas. Durante a execução do algoritmo, os indivíduos são modificados utilizando-se os operadores de mutação e recombinação gênica, a fim de se obterem melhores alinhamentos ao longo das iterações.

Assim, pode-se encontrar diversas implementações bem sucedidas de AG em diferentes ferramentas da área, como a SAGA (NOTREDAME; HIGGINS, 1996) e a MSA-GA (GONDRO; KINGHORN, 2007). Estas ferramentas obtiveram resultados significativos, visto que em muitos casos são capazes de produzir melhores alinhamentos quando comparadas a outras ferramentas comumente utilizadas em Bioinformática, como a ClustalW (GONDRO; KINGHORN, 2007).

A SAGA foi o trabalho pioneiro em implementação de AG para alinhamentos múltiplos de sequências, utilizando-se de uma estratégia híbrida de programação dinâmica e Algoritmo Genético (NOTREDAME; HIGGINS, 1996). Além disso, a SAGA conta com um complexo pacote de 22 operadores genéticos e um escalonador que seleciona qual o melhor operador a ser utilizado em uma determinada ocasião, para a produção dos alinhamentos (NOTREDAME; HIGGINS, 1996).

No entanto, Thomsen e Boomsma evidenciaram que a utilização desta grande quantidade de operadores não proporciona melhores resultados quando comparados aos alinhamentos obtidos com o uso de um seletor de operadores uniforme de um AG simples (THOMSEN; BOOMSMA, 2004). Eles também mostraram que os operadores de mutação contribuem mais que os operadores de recombinação para a obtenção de melhores alinhamentos.

Além disso, segundo Lee, devido à natureza dos Algoritmos Genéticos, pode-se facilmente se esbarrar num ponto de máximo local, o que se apresenta como um problema em alinhamentos de sequências (LEE et al., 2008).

Com isso, observa-se na literatura específica da área que diversas pesquisas continuam colaborando para a constante melhoria dos alinhamentos obtidos através da

heurística de AG, em que esforços especiais têm sido voltados para o refinamento dos operadores genéticos.

Chowdhury e Garai desenvolveram uma abordagem que utiliza operadores de recombinação controlados e mutação guiada para a obtenção de melhores resultados (CHOWDHURY; GARAI, 2014). Amorim implementou a função objetivo COFFEE (*Consistency based Objective Function For alignmEnt Evaluation*) junto à ferramenta MSA-GA e observou resultados com maior significância biológica em alinhamentos de conjuntos de sequências pouco similares (AMORIM et al., 2015).

Além disso, atualmente encontram-se diversas abordagens que implementam estratégias de funções multiobjetivo, o que permite a otimização das soluções considerando-se simultaneamente vários pontos importantes para o alinhamento. Kaya implementou uma estratégia de algoritmos genéticos com função multi-objetivo e obteve melhores resultados quando comparados a outras ferramentas para alinhamento múltiplo de sequências (KAYA; SARHAN; ALHAJJ, 2014), visto que a estratégia implementada permite que o algoritmo convirja para uma solução otimizada por mais de uma característica desejada. De maneira similar, Zhu e He implementaram um algoritmo genético com função multi-objetivo baseada em decomposição, de modo que o algoritmo de alinhamento é divido em duas etapas. Na primeira etapa, a ferramenta busca por um conjunto de pré soluções, enquanto na segunda, o conjunto previamente obtido é processado para a obtenção de um nível de refinamento ainda maior (ZHU; HE; JIA, 2015). Recentemente, Kaya modificou seu modelo de função multiobjetivo, de modo a permitir a obtenção de diferentes alinhamentos em uma única execução do algoritmo (KAYA; KAYA; ALHAJJ, 2016). Essa abordagem leva em consideração dois objetivos conflitantes: minimização do tamanho do alinhamento produzido e maximização da similaridade. Os resultados obtidos por Kaya evidenciaram, na maioria dos casos, uma melhoria na precisão dos alinhamentos produzidos em termos de qualidade biológica (KAYA; KAYA; ALHAJJ, 2016).

Assim, percebe-se que a heurística de AG voltada para problemas de alinhamentos múltiplos de sequências continua com um forte apelo científico na área de Bioinformática.

2.4.5 Otimização por Colônia de Formigas

A heurística de Otimização por Colônia de Formigas é uma técnica bioinspirada comumente utilizada para a resolução de problemas de otimização e busca de caminho mínimo (DORIGO; BLUM, 2005; DORIGO; GAMBARDELLA, 1997).

A grande vantagem da implementação do modelo OCF é o seu baixo custo computacional, em termos de tempo de execução, aliado ao seu modelo matemático consistente, o que geralmente possibilita a obtenção de bons resultados em um tempo hábil (PAPADIMITRIOU; STEIGLITZ, 1998).

Assim, em termos de abstração e modelagem computacional, busca-se entender a maneira como as formigas estabelecem o melhor caminho, partindo do ninho, até uma fonte de comida (DORIGO; GAMBARDELLA, 1997), conforme pode ser visualizado na Figura 2.7.

Na natureza, sabe-se que as formigas se comunicam de maneira sensorial, através do feromônio. Assim, quando uma formiga sai do ninho em busca de uma fonte de comida, ela deposita uma determinada quantidade de feromônio no caminho percorrido, de modo que quanto mais formigas passarem por aquele local, maior será a quantidade de feromônio depositado (DORIGO; GAMBARDELLA, 1997).

Assim, conforme o exemplo da Figura 2.7, observa-se que as Formigas 1, 2 e 3 saíram do ninho em busca de comida, deixando uma determinada quantidade de feromônio nos caminhos 1, 2 e 3, respectivamente. Neste contexto, enquanto as Formigas 2 e 3 continuam procurando por comida, a Formiga 1 encontra a fonte e retorna ao ninho utilizando o mesmo caminho que percorreu até a fonte de comida. Como a Formiga 1 utilizou a referida trilha duas vezes (do ninho à comida e da comida ao ninho), a quan-

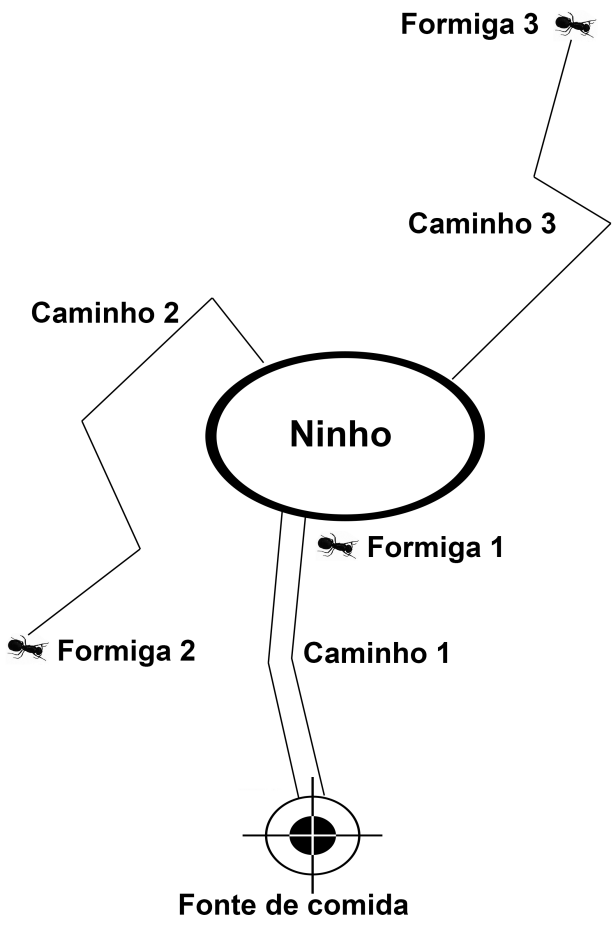


Figura 2.7: Esquema da busca por comida em uma colônia de formigas.

tidade de feromônio depositada naquele caminho é duas vezes maior que nos outros trajetos considerados no exemplo. Desta forma, a probabilidade de outras formigas utilizaram o caminho 1 em busca de uma fonte de comida é muito maior (ZAFALON, 2009).

Sob o ponto de vista computacional, cada caminho representa uma possível solução ótima. Consequentemente, se determinado resultado for de fato ótimo, a quantidade de feromônio naquela trilha será maior e, assim, mais formigas irão seguir este caminho. Isto faz com que o algoritmo estabeleça uma convergência para um ponto ótimo de solução que contenha as características desejadas (BLUM; ROLI, 2003; MOSS; JOHNSON, 2003).

Em Bioinformática, observa-se a aplicação da heurística de OCF em diversos contextos, mas, principalmente, para a resolução de problemas de alinhamentos múltiplos de sequências. Nestes casos, o modelo de Colônia de Formigas pode ser utilizado com o intuito de acelerar a busca por *motifs* em biossequências, os quais podem representar regiões de interesse nos organismos analisados (MOSS; JOHNSON, 2003).

Atualmente, diversas pesquisas têm conduzido a aplicação da OCF em problemas de Bioinformática. SaikatKar desenvolveu uma abordagem baseada em Colônia de Formigas para alinhamentos múltiplos de longas sequências de DNA, o que possibilitou, neste caso, a obtenção de bons resultados em tempo factível (SAIKATKAR; SAHA; CHAKRABARTI, 2014). Além disso, observa-se também a aplicação da OCF para auxiliar na predição de estruturas tridimensionais de proteínas, mais precisamente na fase inicial do processo de enovelamento, que é uma etapa computacionalmente dispendiosa (LV et al., 2013).

Uma outra tendência notável na área de Bioinformática é a implementação da OCF junto a outras heurísticas, a fim de amenizar possíveis deficiências de cada estratégia, como o problema de máximo local nos Algoritmos Genéticos (LEE; KING, 2014).

2.4.6 Heurísticas Híbridas

Conforme elucidado nas subseções anteriores, geralmente a aplicação de heurísticas na resolução de problemas de otimização possui pontos positivos e negativos, ressaltando-se que para cada caso, deve-se analisar qual estratégia melhor se encaixa na resolução do problema considerado.

No entanto, nota-se uma disposição muito grande de pesquisadores de diferentes áreas em combinar diversas heurísticas, de modo a amenizar os possíveis pontos negativos das técnicas empregadas. Vasant aplicou a hibridização das heurísticas de Busca Tabu e lógica *Fuzzy* no tratamento de problemas de planejamento de produção em refinarias (VASANT; GANESAN; ELAMVAZUTHI, 2012). Hoseini e Shayes-

teh combinaram as heurísticas de OCF, AG e Recozimento Simulado para melhorar o contraste em imagens digitais de maneira mais eficiente (HOSEINI; SHAYESTEH, 2013).

Em Bioinformática, observa-se diversos trabalhos que utilizam heurísticas híbridas nas mais diversas aplicações. Shen implementou a técnica de Otimização de Enxames combinada com a heurística de Busca Tabu para a seleção de genes e classificação de tumores (SHEN; SHI; KONG, 2008). Keller aplicou heurísticas de alinhamentos múltiplos de sequências na obtenção de assinaturas para posteriores inferências nas predições de genes. Neste caso, a partir dos resultados obtidos, verificou-se que com a hibridização os alinhamentos apresentaram melhor qualidade (KELLER et al., 2011). De maneira similar, Pierri utilizou estratégias de alinhamentos múltiplos de sequências em combinação com triagens de encaixe molecular para a predição de funções de proteínas desconhecidas (PIERRI; PARISI; PORCELLI, 2010).

Além disso, percebe-se que a abordagem de hibridização de heurísticas possui um forte apelo principalmente na vertente de alinhamentos de sequências. Sun implementou a técnica de Otimização de Enxames na fase de treinamento do Modelo Oculto de Markov (HMM), o que proporcionou uma melhoria no tempo de execução e também na qualidade biológica dos alinhamentos múltiplos de sequências produzidos (SUN et al., 2012). Jangam e Chakraborti aplicaram técnicas de AG junto com OCF para alinhamentos de pares de sequências de DNA (JANGAM; CHAKRABORTI, 2007), o que permitiu a obtenção de alinhamentos mais precisos em um tempo de execução menor. Em uma de suas pesquisas, Lee utilizou Algoritmos Genéticos para alinhamentos múltiplos de sequências e aplicou o modelo de Colônia de Formigas para alinhamento local, conforme observa-se no esquema apresentado na Figura 2.8. Basicamente, a OCF é executada, após a definição da nova população de indivíduos, junto às regiões pouco similares das sequências envolvidas. Com esta estratégia possibilitou-se amenizar o problema de máximo local causado pela natureza do AG, além de melhorar a

qualidade do alinhamento em regiões de baixa similaridade, (LEE et al., 2008).

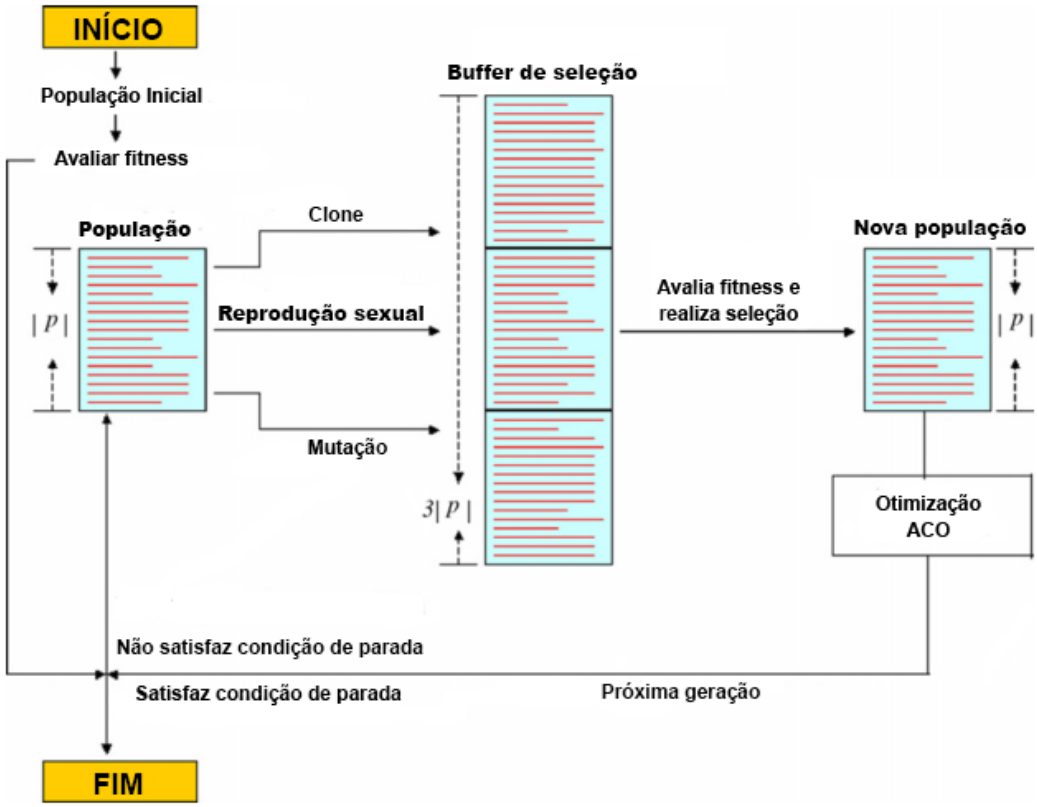


Figura 2.8: Fluxograma da utilização da heurística de AG junto com a OCF. Adaptado de (LEE et al., 2008).

Como outra possibilidade de hibridização de heurísticas, Lee e King utilizaram a OCF para a execução de alinhamentos múltiplos de sequências e aplicaram estratégias de Algoritmo Genético para a otimização dos parâmetros do modelo de Colônia de Formigas, o que permite que sejam associados os melhores valores para cada parâmetro da OCF em cada alinhamento específico (LEE; KING, 2014).

Mais recentemente, Rubio-Largo utilizou um modelo artificial de colônia de abelhas junto com estratégias de alinhamento progressivo para alinhamentos múltiplos de sequências (RUBIO-LARGO; VEGA-RODRÍGUEZ; GONZÁLEZ-ÁLVAREZ, 2016). Basicamente, em meio à execução do algoritmo de colônia de abelhas, utiliza-se a ferramenta *Kalign* para realinhar regiões com concentração de *gaps*. Após este procedimento, reinsere-se a região realinhada ao alinhamento múltiplo e o algoritmo de

colônia de abelhas continua a sua execução. Com esta abordagem, Rubio-Largo obteve resultados superiores aos de ferramentas muito utilizadas em Bioinformática (RUBIO-LARGO; VEGA-RODRÍGUEZ; GONZÁLEZ-ÁLVAREZ, 2016).

Portanto, observa-se que atualmente a implementação de heurísticas híbridas encontra-se em destaque em diversas frentes de pesquisa, principalmente na Bioinformática.

2.5 Ferramentas para Alinhamentos Múltiplos de Sequências

Conforme ressaltado na Seção 2.4, existem diversas heurísticas aplicáveis à resolução de problemas de alinhamentos múltiplos de sequências. Assim, apresentam-se na presente Seção algumas ferramentas frequentemente utilizadas na área de Bioinformática e que fazem uso das heurísticas anteriormente citadas.

2.5.1 Clustal Omega

As ferramentas da família Clustal têm sido comumente utilizadas em Bioinformática, desde a ClustalW (THOMPSON; HIGGINS; GIBSON, 1994). Devido à evolução dos dados e tecnologias de sequenciadores disponíveis, diversas melhorias foram conduzidas a esta ferramenta que atualmente encontra-se na versão conhecida como Clustal Omega (SIEVERS; HIGGINS, 2014).

A Clustal Omega é uma ferramenta para alinhamento múltiplo de sequências que utiliza a heurística de Alinhamento Progressivo como base, assim como nas versões predecessoras da família Clustal. A grande vantagem da Clustal Omega em relação às ferramentas anteriores é que com as melhorias implementadas tornou-se factível o alinhamento múltiplo de sequências envolvendo um volume muito grande de dados, o que anteriormente, de modo geral, degradava a qualidade dos resultados obtidos (SIEVERS et al., 2011). Esta evolução foi necessária, principalmente com o advento

do Sequenciadores de Nova Geração (NGS), pois estes produzem volumes de dados bem superiores aos Sanger até então muito utilizados (GLENN, 2011).

Do ponto de vista da eficiência computacional, a grande vantagem da Clustal Omega é a implementação do algoritmo *mBED* na etapa de construção da árvore filogenética, a qual apresenta-se como uma das mais custosas na heurística de Alinhamento Progressivo. Assim, com as adequações do algoritmo para ambientes sequenciais, proporcionou-se a viabilidade de execução do alinhamento de um grande volume de sequências (SIEVERS et al., 2011).

Além disso, para refinar os resultados obtidos pela Clustal Omega em termos de significância biológica, utilizou-se o algoritmo *HHalign*, o qual baseia-se no HMM. Outros fatores importantes são a possibilidade da reutilização de alinhamentos, o que evita recálculos, e o fato da viabilidade de especificação do perfil do HMM, o qual pode ser derivado de sequências homólogas às de entrada (SIEVERS et al., 2011).

No entanto, conforme apontado por Boyce, ferramentas baseadas em Alinhamento Progressivo possuem forte dependência da ordem das sequências no conjunto de entrada, o que pode degradar a qualidade final do alinhamento (BOYCE; SIEVERS; HIGGINS, 2015).

2.5.2 DIALIGN

A DIALIGN é uma ferramenta para alinhamentos de sequências de nucleotídeos e aminoácidos, seja utilizando-se de uma estratégia exata para alinhamentos par-a-par ou mesmo de uma abordagem estocástica para alinhamentos múltiplos de sequência (MORGENSTERN et al., 1998). A grande diferença é que esta ferramenta não realiza a comparação de resíduos isolados e também não aplica penalizações por *gaps*, como comumente é feito em outras ferramentas da área.

Basicamente, a DIALIGN utiliza uma estratégia de alinhamento local, o que privilegia a busca por determinados *motifs* em biossequências, os quais podem representar

importantes pontos de conservação dos organismos analisados (MORGENSTERN et al., 1998).

Para a construção do alinhamento propriamente dito, a DIALIGN utiliza um esquema de comparação em diagonal das sequências envolvidas, conforme observa-se na Figura 2.9.

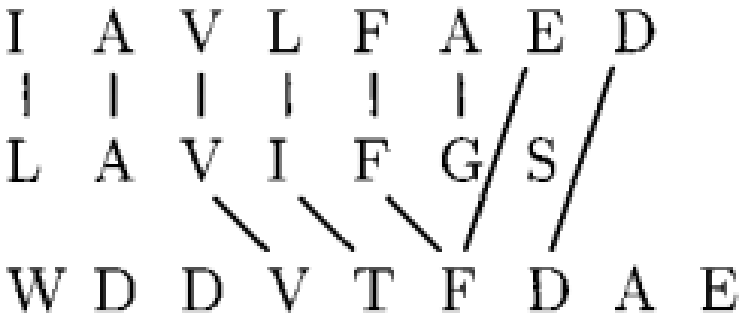


Figura 2.9: Cálculo de diagonal no DIALIGN. Fonte: (MORGENSTERN et al., 1998).

Os pares de sequências são comparados em diagonal para a construção de uma matriz de pontuação, a fim de se obter um conjunto de diagonais que satisfaçam um determinado critério de consistência. As diagonais obtidas são ponderadas segundo um fator probabilístico, em que se busca por uma conexão entre as diagonais de modo a maximizar a soma dos pesos que foram atribuídos (MORGENSTERN et al., 1998).

Embora a versão inicial da DIALIGN possua a capacidade de produzir bons alinhamentos, o custo computacional, em termos de tempo de execução, é alto. Deste modo, Schmollinger utilizou uma abordagem paralela para amenizar o problema do tempo de execução apresentado pela ferramenta (SCHMOLLINGER et al., 2004).

Além disso, estudos apontaram que a natureza do algoritmo utilizado na DIALIGN pode conduzir o alinhamento final a um resultado subótimo, o que geralmente é indesejável (SUBRAMANIAN et al., 2008). Com isso, Subramanian implementou uma estratégia que combina as abordagens previamente utilizadas com Algoritmo Progressivo (SUBRAMANIAN et al., 2008), o que possibilitou amenizar o problema ocorrido devido à natureza do algoritmo utilizado na DIALIGN.

2.5.3 MAFFT

Uma ferramenta consolidada no contexto da Bioinformática é a MAFFT (*Multiple Alignment using Fast Fourier Transform*) (KATOH et al., 2002). Esta ferramenta utiliza a transformada rápida de Fourier (FFT) para a execução de alinhamentos múltiplos de sequências de nucleotídeos ou aminoácidos.

A grande vantagem proporcionada pela MAFFT foi a obtenção de alinhamentos com boa significância biológica em um tempo de processamento consideravelmente menor que as abordagens existentes (KATOH et al., 2002).

Basicamente, o algoritmo de alinhamento da MAFFT divide-se em duas etapas principais: a primeira visa a detecção de regiões homólogas nas sequências de entrada por meio da FFT, enquanto a segunda oferece um método eficiente para o cálculo da pontuação do alinhamento (KATOH et al., 2002). É importante ressaltar que a MAFFT tem sido utilizada em diversas pesquisas que objetivam a evolução dos processos de alinhamento e, portanto, encontra-se atualmente na sua versão 7 (KATOH; STANDLEY, 2013). Mas, recentemente, recebeu novas alterações e funcionalidades, como um ambiente de interface web (KATOH; STANDLEY, 2014).

Além disso, assim como outras ferramentas da área, houve a necessidade de adequar a MAFFT à nova era de dados intensivos em que se encontra a Bioinformática. Deste modo, Katoh e Toh adaptaram a ferramenta ao paradigma *multithreading*, o que possibilitou a execução dos processos nos diferentes núcleos do processador de uma única máquina (KATOH; TOH, 2010). Mais recentemente, Zhu implementou uma abordagem paralela da ferramenta utilizando GPUs (*Graphics Processing Unity*) (ZHU et al., 2015).

No entanto, ainda percebe-se que a MAFFT encontra uma certa deficiência na tentativa de realizar alinhamentos múltiplos de sequências para conjuntos pouco relacionados ou conjuntos cujas sequências possuam vários trechos de inserção (KATOH; STANDLEY, 2013).

A fim de amenizar as desvantagens da MAFFT nesses casos, Katoh e Standley implementaram uma estratégia com base em uma matriz de pontuação variável, de modo a oferecer maior precisão ao alinhar regiões pouco relacionadas (KATOH; STANDLEY, 2016). Embora melhorias tenham sido obtidas, a estratégia implementada ainda possui limitações, como o fato de estimar a distância de sequências não homólogas (KATOH; STANDLEY, 2016).

2.5.4 SAGA

Conforme citado na Subseção 2.4.4, a ferramenta SAGA foi a pioneira ao implementar a heurística de Algoritmo Genético para alinhamentos múltiplos de sequências, em uma abordagem híbrida que também utiliza programação dinâmica como parte do processo (NOTREDAME; HIGGINS, 1996).

Basicamente, a ideia principal da ferramenta, conforme pode ser visto na Figura 2.10, consiste em evoluir a população inicial de alinhamentos, segundo processos de reprodução sexual, até que uma condição de parada seja satisfeita (NOTREDAME; HIGGINS, 1996).

A SAGA é composta por um complexo conjunto de 22 operadores de mutação e recombinação gênica, que são executados de acordo com um escalonador automático que utiliza determinados critérios para decidir qual o melhor operador a ser utilizado em determinada iteração (NOTREDAME; HIGGINS, 1996).

No entanto, posteriormente Thomsen e Boomsa evidenciaram que tal complexidade é desnecessária e que o escalonador automático da SAGA não proporciona melhorias aos alinhamentos quando comparados aos resultados obtidos por ferramentas que utilizam Algoritmos Genéticos simples com base em um seletor uniforme de operadores (THOMSEN; BOOMSMA, 2004).

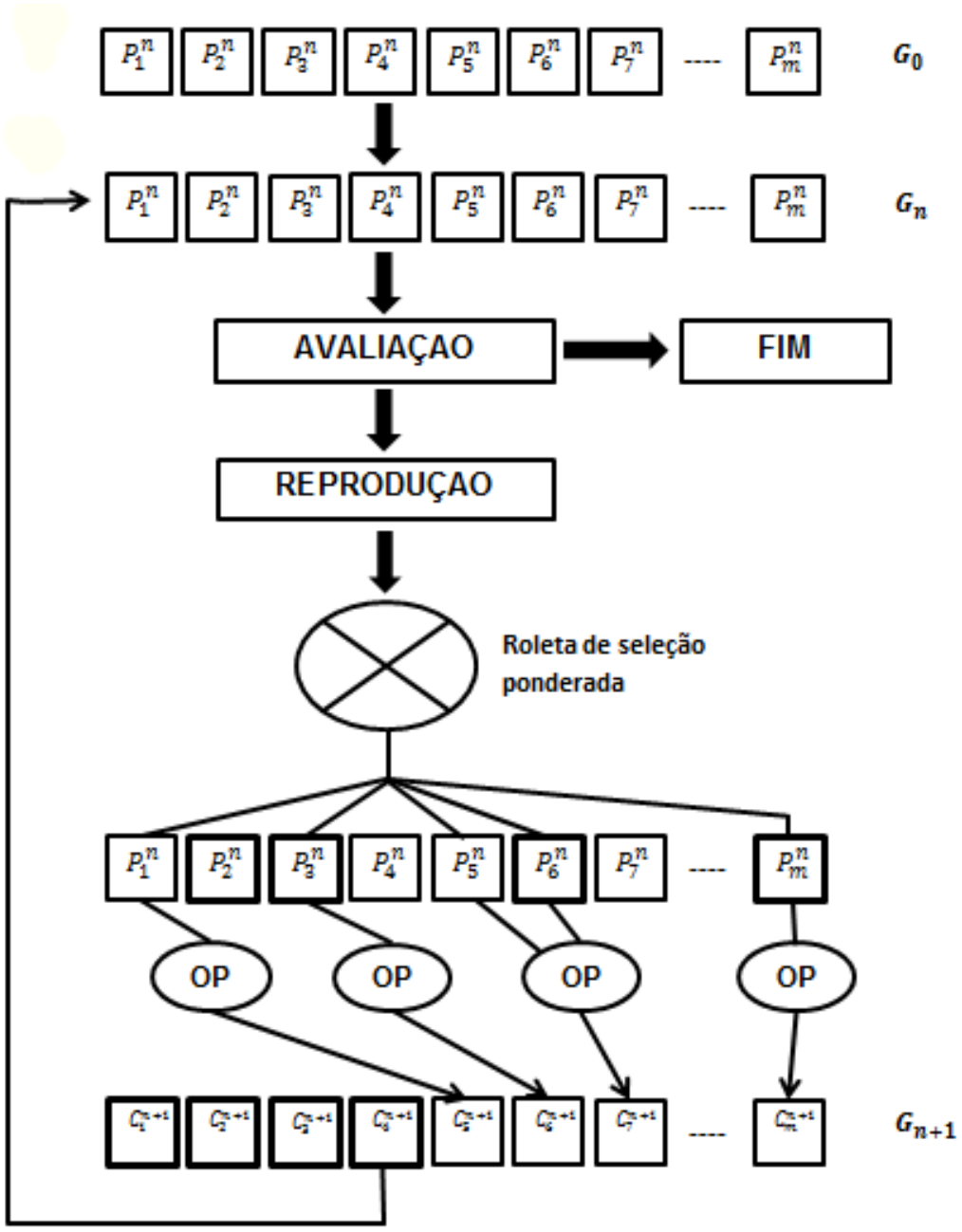


Figura 2.10: Esquema do algoritmo implementado na SAGA. Adaptado de: (NOTRE-DAME; HIGGINS, 1996).

2.5.5 MSA-GA

A MSA-GA (GONDRO; KINGHORN, 2007) trata-se de uma ferramenta para alinhamentos múltiplos de sequências de nucleotídeos ou aminoácidos que, assim como a SAGA, utiliza a heurística de AG como base. Em contrapartida, a MSA-GA im-

plementa uma abordagem simples que faz uso de um seletor uniforme de operadores genéticos, ao contrário da complexa abordagem contemplada na SAGA.

Na MSA-GA, representa-se um alinhamento por meio de uma matriz de n linhas por L colunas, em que cada elemento da matriz é uma sequência do grupo de entrada a ser alinhado. Em termos de representação, quando trata-se de uma sequência de nucleotídeos, cada posição no vetor é ocupada por um caracter que pertence ao alfabeto {A, T, C, G, -}, cujos componentes representam a Adenina, a Timina, a Citosina, a Guanina e uma posição de *gap*, respectivamente. De maneira similar, quando o conjunto de entrada é composto por sequências de proteínas, cada posição do vetor é ocupado por algum elemento do alfabeto de aminoácidos, cujas abreviações encontram-se representadas na Tabela 2.1, na Subseção 2.2.2.

Assim, o número de colunas da matriz é calculado segundo a Equação 2.1 (GONDRO; KINGHORN, 2007), em que L representa o tamanho da maior sequência multiplicado por um fator k e, por fim, soma-se um valor x que representa o deslocamento inicial definido pelo usuário.

$$L = max(l_1, l_2, ..., l_n) * k + x \tag{2.1}$$

Na MSA-GA cada indivíduo da população representa uma possível solução. Com isso, a ferramenta faz uso de um algoritmo para preencher a população inicial de candidatos, conforme pode ser observado na Figura 2.11.

Basicamente, a MSA-GA realiza o alinhamento par-a-par das sequências de entrada, por meio do algoritmo de Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970). Após este processo, seleciona-se aleatoriamente uma sequência correspondente a cada par previamente alinhado, de modo a compor o organismo inicial. Além disso, adiciona-se uma quantidade aleatória de *gaps* nas posições iniciais das sequências selecionadas, restringindo-se ao valor máximo do deslocamento inicial previamente definido.

```
n = 1
for i = 1 to NumSeq – 1
        for j = i+1 to NumSeq
                PairAlign( ) = DP_Align(i,j);
                Alignments(i,n) = PairAlign(1);
                Alignments(j,n) = PairAlign(2);
        n++;
for i = 1 to PopSize
        for j = 1 to NumSeq
                Population(i,j) = InitGaps(Random(MaxOfsset))+Alignments(j,Random(NumSeq-1));
        Fitness(i) = Calc_Fitness(i);
```

Figura 2.11: Algoritmo para preenchimento da população inicial da MSA-GA. Fonte: (GONDRO; KINGHORN, 2007).

Uma vez que a população inicial do AG tenha sido definida, cada indivíduo é então avaliado por alguma função objetivo que reflita o nível de similaridade entre as sequências do alinhamento em questão (GONDRO; KINGHORN, 2007). A função objetivo padrão da MSA-GA é a bem conhecida *Weighted Sum-of-Pairs* (WSP) , a qual encontra-se representada pela Equação 2.2. Basicamente, a pontuação de cada par de resíduos é dada segundo alguma matriz de substituição que se encontra implementada na MSA-GA. Além disso, esta pontuação é ponderada por um fator w, o qual pode ser definido pelo usuário ou mesmo derivado das pontuações dos alinhamentos par-a-par (GONDRO; KINGHORN, 2007).

$$Weighted\ Sum\ of\ Pairs = \sum_{i=2}^{k} \sum_{j=1}^{i-1} w_{ij}\, s_{ij} \tag{2.2}$$

No entanto, Notredame observou que a função WSP possui uma limitação ao avaliar conjuntos de sequências pouco similares, o que pode resultar em alinhamentos com significância biológica indesejável (NOTREDAME; HOLM; HIGGINS, 1998).

Assim, Amorim implementou a função objetivo COFFEE junto à MSA-GA a fim de amenizar o problema causado pela WSP (AMORIM et al., 2015).

Basicamente, a função COFFEE, cujo esquema encontra-se representado na Figura 2.12, baseia-se no critério de consistência entre as sequências a serem alinhadas. De forma geral, a função objetivo contribui para a convergência do AG para uma solução cujas sequências apresentem similaridade máxima com os alinhamentos par-a-par da biblioteca, os quais foram obtidos por meio de programação dinâmica.

Após a avaliação dos indivíduos da população inicial, estes são expostos aos operadores de mutação e recombinação gênica, a fim de evoluir os candidatos a alinhamento (GONDRO; KINGHORN, 2007).

Pelo fato da MSA-GA implementar uma abordagem simples de AG, observa-se que a ferramenta possui um número reduzido de operadores de recombinação e mutação gênica. Considerando-se os operadores de recombinação, tem-se a Recombinação Horizontal e a Recombinação Vertical, as quais encontram-se representadas na Figura 2.13. Na Recombinação Horizontal, o operador seleciona aleatoriamente as sequências dos pais, de modo a combiná-las para gerar um novo indivíduo. Na Recombinação Vertical, o operador define um ponto de corte nas sequências dos pais e une os resíduos do primeiro indivíduo, a partir da posição inicial até o ponto de corte, com os resíduos do segundo pai, a partir do ponto de corte até o fim das sequências. Independente do tipo de operador de recombinação, o processo consiste na seleção de dois indivíduos que, por meio das estratégias supracitadas, geram um novo filho.

Além dos processos de recombinação, a MSA-GA implementa alguns operadores de mutação gênica, os quais atuam apenas com operações que envolvem posições de *gap*. Assim, pode-se abrir um novo bloco de *gaps* em determinada posição do alinhamento, fechar um bloco de *gaps* existente e ampliar ou reduzir o tamanho de um bloco de *gaps*.

Uma vez que os novos indivíduos foram gerados, uma nova população é criada.

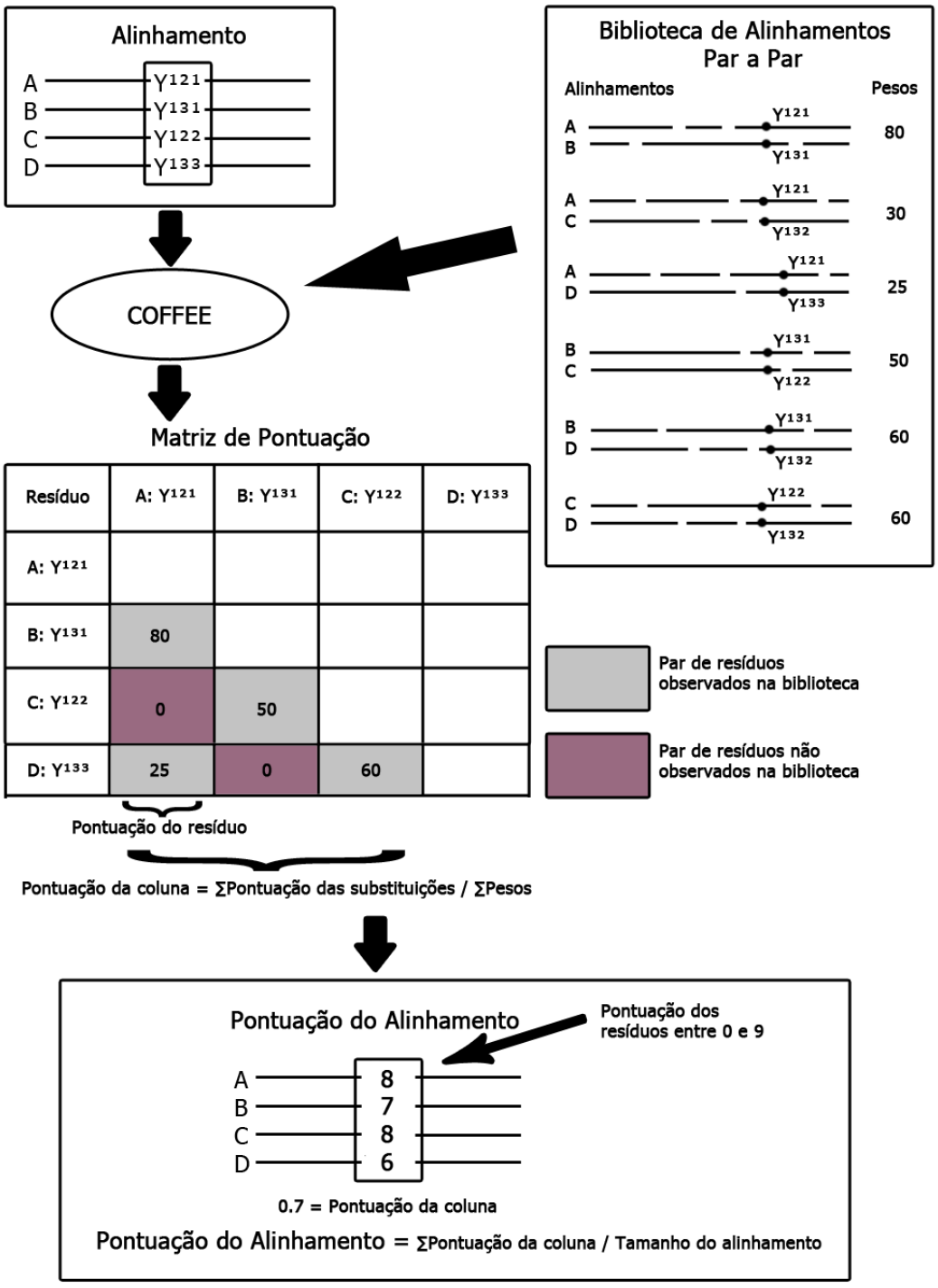


Figura 2.12: Esquema da função objetivo COFFEE. Adaptado de: (NOTREDAME; HOLM; HIGGINS, 1998).

Com isso, a MSA-GA avalia novamente todos os indivíduos e o algoritmo é iterado até que o número de gerações definido pelo usuário seja alcançado. Ao término do

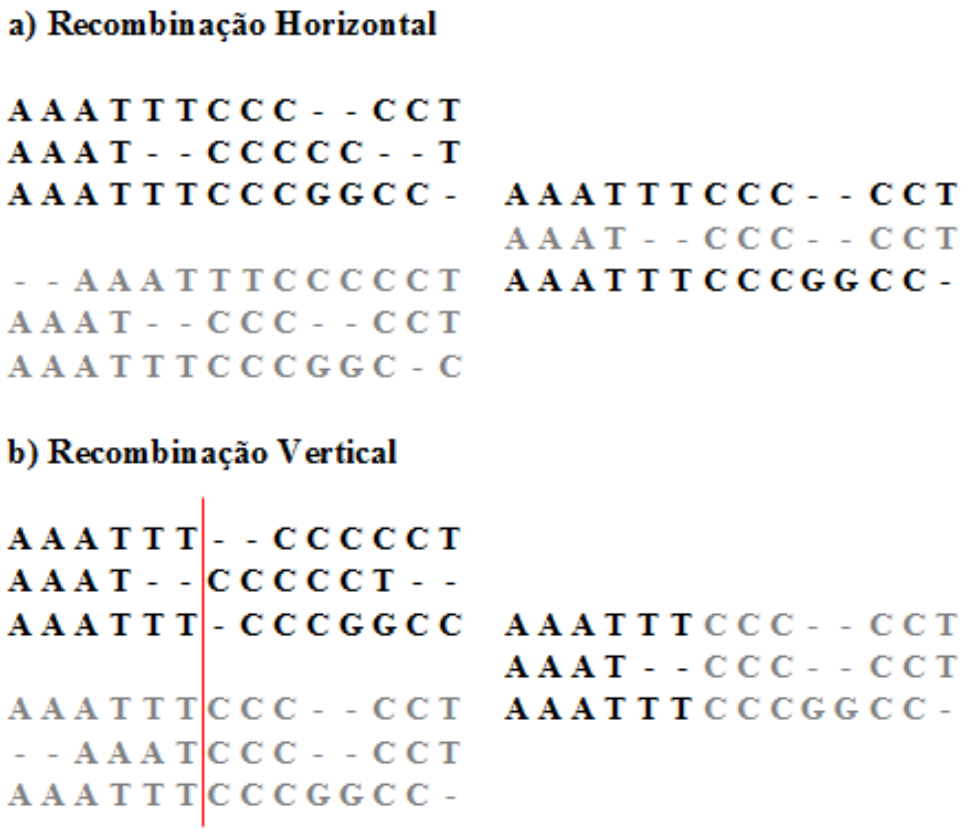


Figura 2.13: Operadores de a) Recombinação Horizontal e b) Recombinação Vertical. Adaptado de: (GONDRO; KINGHORN, 2007).

processo, o indivíduo com maior pontuação será considerado o melhor alinhamento encontrado pela ferramenta.

2.6 Considerações Finais

Neste capítulo foram apresentados os conceitos fundamentais sobre biologia molecular que são importantes para o entendimento do presente trabalho. Além disso, descreveu-se as diferentes heurísticas para alinhamentos múltiplos de sequências e qual o estado da arte em que se encontram cada uma delas. Por fim, apresentou-se o ferramental existente para o problema de alinhamento múltiplo de sequências.

Capítulo 3

Desenvolvimento do Trabalho

3.1 Considerações Iniciais

Este capítulo tem o objetivo de apresentar os detalhes do desenvolvimento do trabalho no que se diz respeito à implementação do escalonador de funções objetivo e à etapa de pós processamento com OCF, junto à ferramenta MSA-GA.

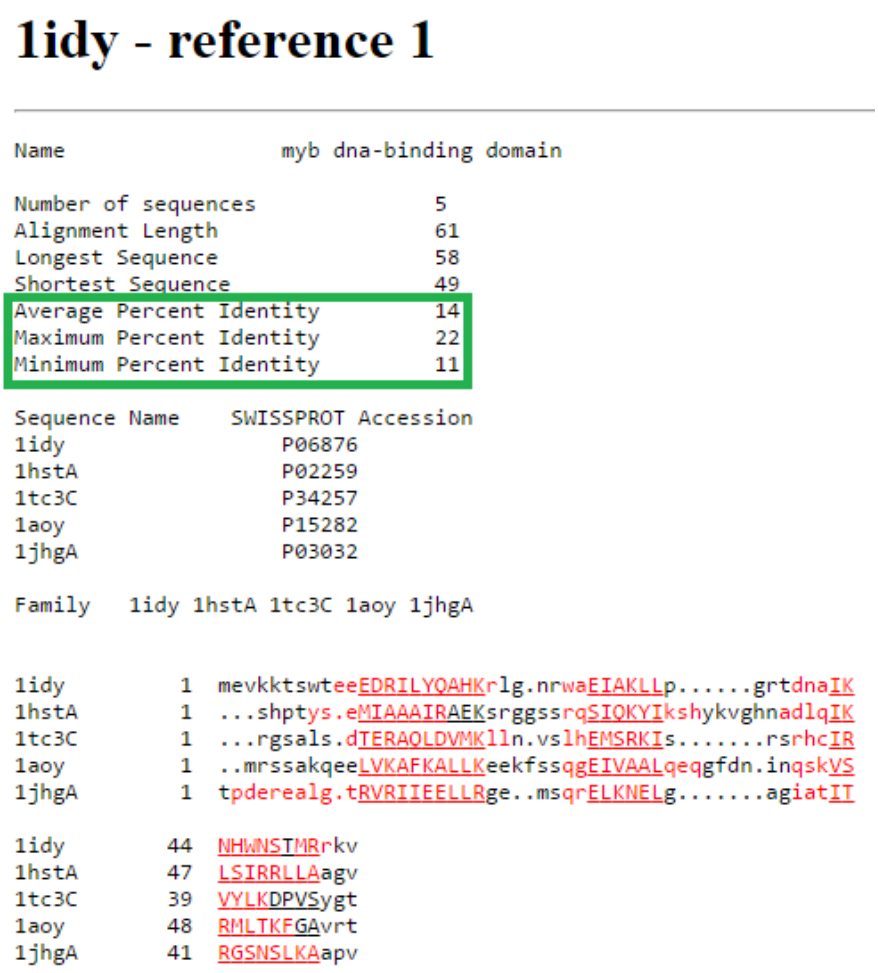
3.2 Escalonador Automático Multifunção

O propósito de se implementar um escalonador automático de funções objetivo na MSA-GA é fazer com que a ferramenta, durante a execução, selecione qual a função mais adequada, entre a COFFEE e a WSP, para cada alinhamento múltiplo específico.

O conceito utilizado no presente trabalho para selecionar a função objetivo mais pertinente baseia-se na similaridade entre as sequências envolvidas no alinhamento. Segundo Amorim et al. (2015), a função objetivo COFFEE é capaz de avaliar de maneira mais precisa os conjuntos de sequências pouco similares, enquanto a função WSP, geralmente, produz melhores resultados para conjuntos de sequências mais correlatas.

Assim, o principal ponto para o desenvolvimento de um escalonador multifunção

é identificar uma maneira para se realizar o cálculo de similaridade entre as sequências do alinhamento múltiplo que será produzido. Para tal, no presente trabalho foi utilizado o *benchmark* BAliBase[1], o qual apresenta conjuntos de sequências manualmente alinhados, além de oferecer uma análise destes alinhamentos, bem como o percentual máximo, mínimo e médio de similaridade para cada caso disponível (THOMPSON et al., 2005), conforme observa-se na Figura 3.1.



1idy - reference 1

```
Name                    myb dna-binding domain

Number of sequences              5
Alignment Length                 61
Longest Sequence                 58
Shortest Sequence                49
Average Percent Identity         14
Maximum Percent Identity         22
Minimum Percent Identity         11

Sequence Name    SWISSPROT Accession
1idy                 P06876
1hstA                P02259
1tc3C                P34257
1aoy                 P15282
1jhgA                P03032

Family   1idy 1hstA 1tc3C 1aoy 1jhgA


1idy         1  mevkktswteeEDRILYQAHKrlg.nrwaEIAKLLp......grtdnaIK
1hstA        1  ...shptys.eMIAAAIRAEKsrggssrqSIQKYIkshykvghnadlqIK
1tc3C        1  ...rgsals.dTERAQLDVMKlln.vslhEMSRKIs.......rsrhcIR
1aoy         1  ..mrssakqeeLVKAFKALLKeekfssqgEIVAALqeqgfdn.inqskVS
1jhgA        1  tpderealg.tRVRIIEELLRge..msqrELKNELg.......agiatIT

1idy        44  NHWNSTMRrkv
1hstA       47  LSIRRLLAagv
1tc3C       39  VYLKDPVSygt
1aoy        48  RMLTKFGAvrt
1jhgA       41  RGSNSLKAapv
```

Figura 3.1: Dados do conjunto de sequências *1idy* junto ao *BAliBase*.

Ao analisar os dados disponíveis, percebe-se que esses percentuais de similaridade se referem aos valores observados nos alinhamentos par-a-par das sequências envolvidas. O percentual de similaridade trata-se da quantidade de correspondências

[1] http://www.lbgi.fr/balibase/

verificadas no alinhamento par-a-par, dividida pelo tamanho do alinhamento em questão. Assim, o percentual de similaridade máximo trata-se do maior valor obtido entre os pares analisados, o percentual mínimo trata-se do menor valor observado, enquanto o percentual médio de similaridade trata-se da média dos percentuais obtidos em todos os pares de sequências.

Deste modo, o percentual médio de similaridade pode ser formalizado pela Equação 3.1, em que $match(i,j)$ é a quantidade de correspondências entre as sequências i e j, $tam(i,j)$ é o tamanho do alinhamento entre as sequências i e j e n é o número de sequências envolvidas no alinhamento múltiplo.

$$sim = \frac{\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{match(i,j)}{tam(i,j)}\right)}{\frac{n*(n-1)}{2}} \tag{3.1}$$

Basicamente, o BAliBase divide os conjuntos de sequências em dois grandes grupos, com base na quantidade de posições de correspondência de resíduos no alinhamento: conjuntos de sequências menos similares ($\leq$35% de similaridade) e mais similares ($>$35% de similaridade). No presente trabalho, utiliza-se o percentual médio de similaridade de cada conjunto de sequências para classificá-los como conjuntos menos ou mais relacionados.

Assim, na Figura 3.2 apresenta-se o fluxograma do escalonador de funções que foi implementado no presente trabalho. Basicamente, pode-se dividir o funcionamento do escalonador de funções em duas etapas: cálculo do percentual médio de similaridade e seleção da função objetivo mais apropriada.

Na primeira etapa, são utilizados os alinhamentos par-a-par computados inicialmente pela MSA-GA, por meio do algoritmo de Needleman-Wunsch. Para cada alinhamento, percorrem-se as sequências em questão, de modo a computar a quantidade de correspondências entre o alinhamento par-a-par analisado. Uma vez que o alinhamento foi percorrido até o fim, divide-se o valor obtido pelo tamanho do alinhamento par-a-par em questão. Assim, com base no par de sequências analisado, obtém-se o

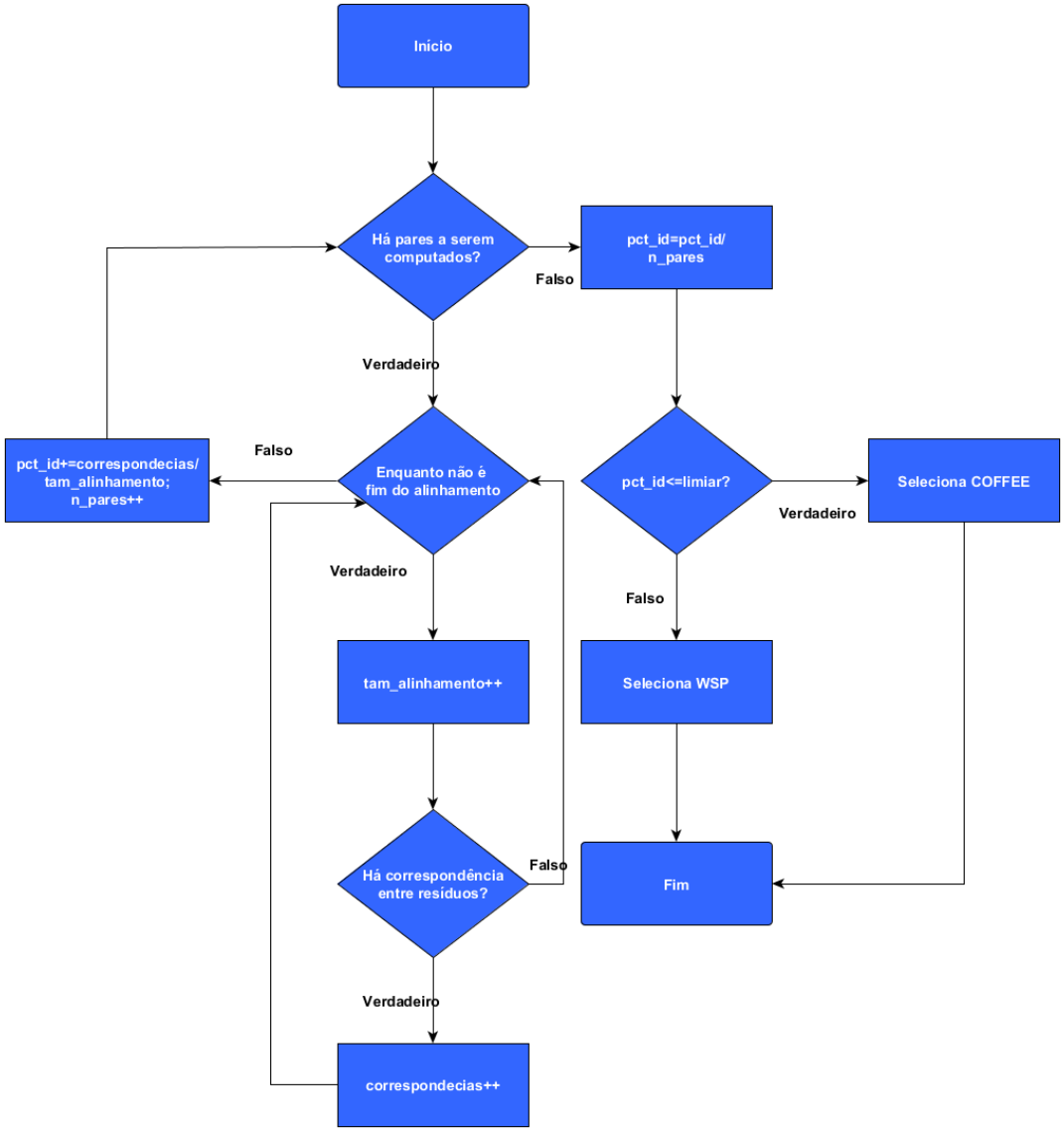


Figura 3.2: Fluxograma do escalonador automático de funções objetivo.

valor do percentual de identidade, o qual é atribuído a uma variável que corresponde ao valor da média das identidades. Uma vez que este processo é executado para todos os pares de sequências, divide-se o valor da soma de todos os percentuais de identidade pela quantidade de alinhamentos par-a-par analisados, o que corresponde, finalmente, ao valor do percentual médio de identidade que será utilizado pelo escalonador para selecionar qual a função objetivo mais adequada.

Assim que o percentual médio de similaridade é obtido, faz-se uma chamada à

função principal do escalonador automático, a qual utiliza este valor para executar entre a função COFFEE e a WSP. Conforme citado anteriormente, os conjuntos de sequências podem ser divididos em grupos mais similares ou menos relacionados. Ao observar a separação feita pelo BAliBase, recomenda-se um limiar de 35% para dividir os conjuntos de sequências analisados. No entanto, ressalta-se que este valor é totalmente parametrizável na estratégia implementada no presente trabalho, conforme observa-se na Figura 3.3.

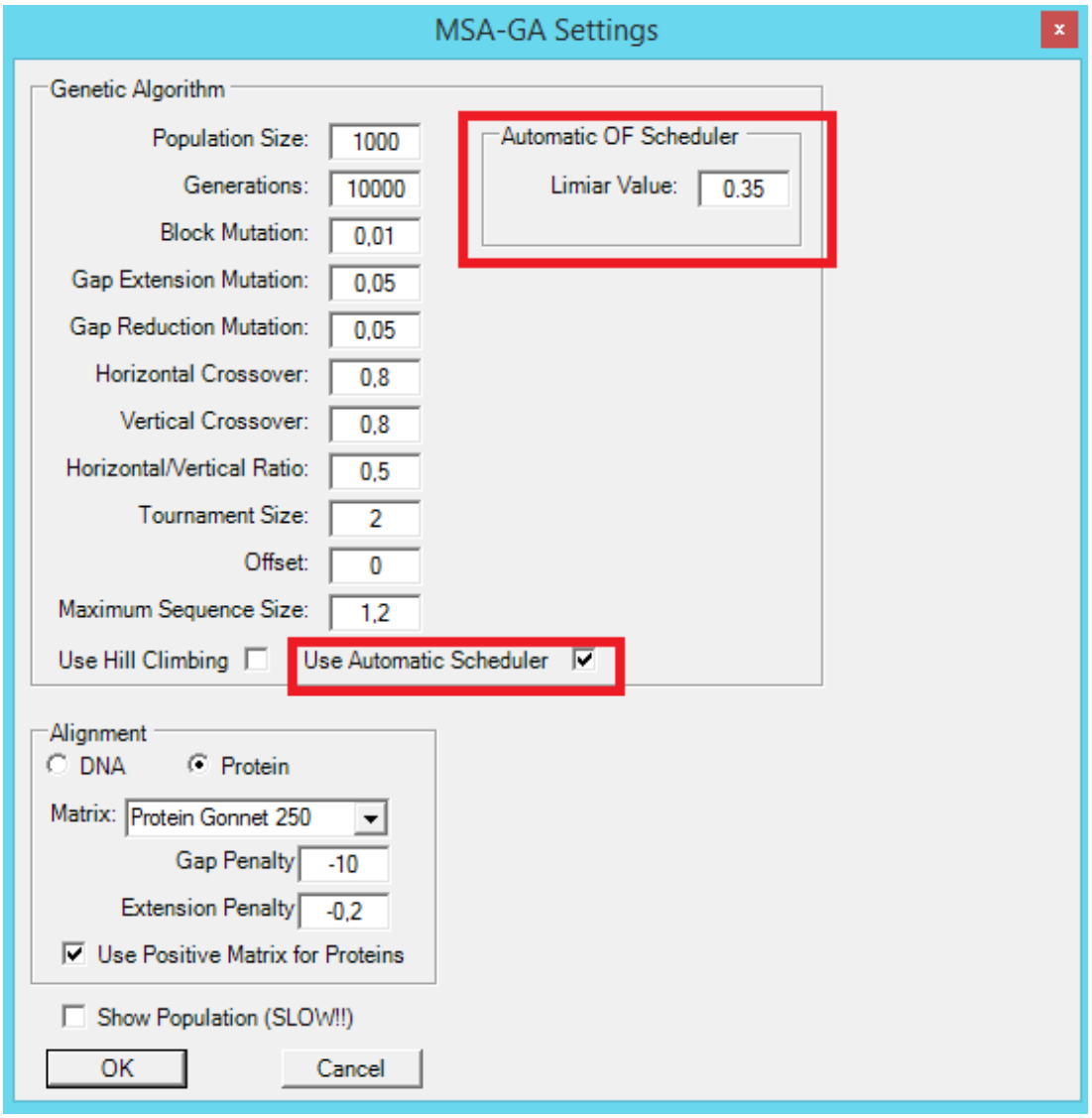


Figura 3.3: Parâmetros da ferramenta MSA-GA.

Com isso, caso a utilização do escalonador automático esteja selecionada, se o per-

centual de similaridade for menor ou igual ao valor do limiar definido pelo usuário, a MSA-GA automaticamente irá selecionar a função objetivo COFFEE como método de avaliação do AG da ferramenta. No entanto, se o percentual de similaridade for maior do que o limiar definido, a função selecionada será a WSP. Caso a opção do escalonador automático esteja desabilitada, a MSA-GA simplesmente seleciona a função objetivo selecionada manualmente pelo usuário, conforme pode ser visto na Figura 3.4.

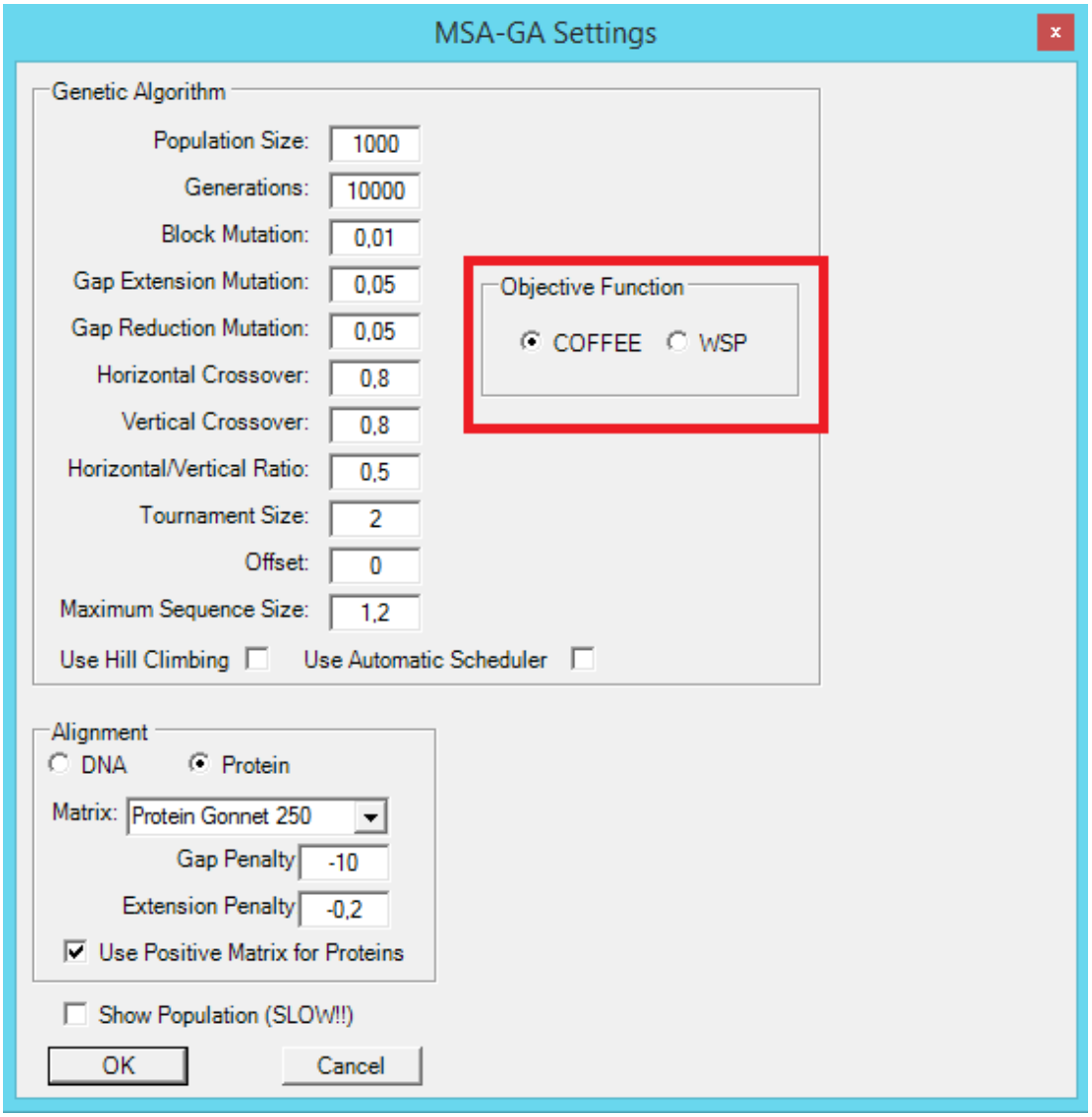


Figura 3.4: Opções para definição manual de funções objetivo junto à MSA-GA.

Assim, com a estratégia implementada no presente trabalho, possibilita-se que a MSA-GA selecione automaticamente a função objetivo mais adequada para cada ali-

nhamento múltiplo específico, com base em informações sobre percentual de similaridade previamente observadas (AMORIM et al., 2015).

3.3 Pós Processamento com Otimização por Colônia de Formigas

Segundo Lee et al. (2008), a heurística de AG pode facilmente se direcionar a um ponto de máximo local, o que significa que, quando aplicada a problemas de alinhamento múltiplo de sequências, pode produzir um resultado que ainda pode ser refinado. Assim, o propósito de se implementar uma estratégia de OCF como uma fase de pós-processamento da MSA-GA é tentar refinar o alinhamento produzido pelo AG da ferramenta. Com isso, o alinhamento gerado a partir do AG é utilizado como entrada para a OCF.

Para se obter a melhoria desejada, foi utilizada no presente trabalho uma adaptação do algoritmo proposto por Zhou et al. (2010), que utiliza OCF para executar alinhamentos par-a-par de biossequências. Esse método se baseia em um algoritmo simplificado de OCF que considera apenas a informação de feromônio para deslocar as formigas em uma estrutura denominada *grid* simplificado, a qual pode ser definida como uma matriz, em que cada célula é conhecida como *grid*. Essa estrutura simula os possíveis caminhos que podem ser seguidos por cada formiga. Assim, dadas duas sequências a serem alinhadas, o algoritmo define uma estrutura baseada nessas informações, conforme pode ser observado na Figura 3.5.

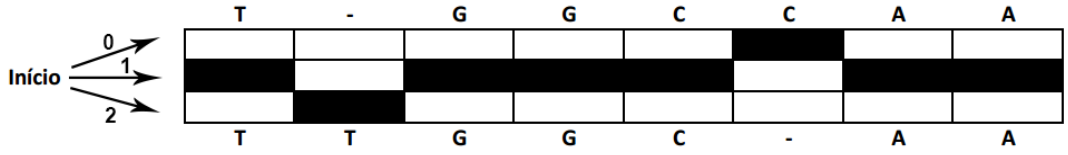


Figura 3.5: Exemplo de alinhamento com a estrutura de *grid* simplificado.

Basicamente, existem sempre três opções de caminhos disponíveis para a formiga,

as quais são o *grid* superior, o *grid* central e o *grid* inferior, descritos pelos números 0, 1 e 2, respectivamente. Quando uma formiga escolhe um *grid*, ela deposita uma determinada quantidade de feromônio que será utilizada futuramente pelo algoritmo. Cada escolha implica em um processamento diferente por parte da OCF, o que será detalhado nas próximas subseções.

Assim, o algoritmo implementado no presente trabalho pode ser dividido em quatro etapas: inicialização da intensidade do feromônio, processamento do deslocamento da formiga, cálculo da pontuação do alinhamento obtido pela formiga e atualização da intensidade do feromônio e, por fim, a montagem do alinhamento múltiplo de sequências.

3.3.1 Inicialização da Intensidade de Feromônio

Para o controle da intensidade de feromônio, o algoritmo utiliza o *grid* simplificado, que nesse caso, trata-se de uma matriz de pontos flutuantes, denominada τ. Inicialmente, deve-se definir o tamanho da estrutura, em que a quantidade de linhas da matriz é sempre igual a três, de modo a representar os três caminhos possíveis (0, 1 e 2) e a quantidade de colunas é determinada por $(2 * maior)$, em que *maior* é o tamanho da maior sequência envolvida no alinhamento.

Uma vez que a estrutura da trilha de feromônio é definida, deve-se preencher cada célula da matriz com a intensidade inicial que será utilizada pelo algoritmo. Essa intensidade é estabelecida por meio de um parâmetro da OCF, chamado τ_0, cujo valor é determinado pelo usuário. Assim, inicialmente, todas as células da estrutura contém a mesma intensidade de feromônio.

3.3.2 Processamento do Deslocamento da Formiga

No algoritmo adotado no presente trabalho, toda escolha de deslocamento das formigas é feita com base na intensidade do feromônio contido nos *grids* da matriz τ.

A posição inicial da formiga na estrutura é a coluna zero, a qual indica o início do alinhamento. A partir disso, a formiga deve selecionar um *grid* para seguir para a próxima coluna do alinhamento. Se a formiga escolher o *grid* superior, definido pelo caminho 0, um *gap* é inserido na sequência inferior, na posição atual, conforme observa-se na Figura 3.6, em que a escolha da formiga é identificada pelo *grid* preenchido.

Figura 3.6: Escolha do *grid* superior.

Analogamente, se a formiga escolher o *grid* inferior, definido pelo caminho 2, um *gap* é inserido na sequência superior, na posição atual, conforme observa-se na Figura 3.7.

Figura 3.7: Escolha do *grid* inferior.

No entanto, se a formiga seleciona o *grid* central, definido pela posição 1, nenhuma posição de *gap* é inserida nas sequências naquela coluna do alinhamento.

Conforme elucidado anteriormente, a regra de transição de uma formiga z se baseia na intensidade do feromônio contido nos *grids*. Assim, para cada *grid* da estrutura, existe uma probabilidade de transição, a qual é calculada com base na Equação 3.2, em que $\tau_{ij}(t)$ representa o feromônio contido no *grid* (i,j), com $i = 0,1,2$ representando os três caminhos possíveis e j a coluna atual do alinhamento na t-ésima iteração; η_{ij}

é uma informação heurística que pondera as bases analisadas, de modo que $\eta_{ij} = 2$ se houver correspondência ou $\eta_{ij} = 1$ caso não haja correspondência. Por fim, α e β são parâmetros que representam os pesos do feromônio, cujos valores são definidos pelo usuário.

$$P_{ij}^{z} = \frac{\tau_{ij}(t)^{\alpha}\eta_{ij}^{\beta}}{\sum \tau_{ij}(t)^{\alpha}\eta_{ij}^{\beta}} \tag{3.2}$$

Assim, todas as probabilidades de transição para os *grids* da coluna do alinhamento que está sendo analisada são calculadas. Além disso, o usuário deve definir um valor para o parâmetro q_0 entre 0 e 1, que representa a probabilidade do algoritmo utilizar as informação previamente calculadas como regra de transição para mover a formiga na estrutura. Posteriormente, a OCF gera aleatoriamente um número q, também entre 0 e 1. Desse modo, se $q \leq q_0$, a formiga irá escolher o *grid* com maior valor de probabilidade de transição. Por outro lado, se $q > q_0$, o algoritmo seleciona, de maneira aleatória, um dos *grids*.

Além disso, é importante ressaltar que independente dos valores das probabilidades de transição dos *grids*, se as bases analisadas forem correspondentes, obrigatoriamente o algoritmo irá selecionar o *grid* central para não inserir posições de *gaps* no alinhamento, de modo a majorar a quantidade de correspondências no alinhamento múltiplo.

Se o algoritmo chegar na última base de uma das sequências, significa que a formiga encontrou um final efetivo. Quando uma formiga chega no final efetivo da sequência inferior, obrigatoriamente o próximo *grid* selecionado será o superior, de modo a inserir uma posição de *gap* na sequência inferior, conforme pode ser visto na Figura 3.8.

Quando uma formiga chega no final efetivo da sequência superior, obrigatoriamente o próximo *grid* selecionado será o inferior, de modo a inserir uma posição de *gap* na sequência superior. Quando uma formiga encontrar um final efetivo para ambas

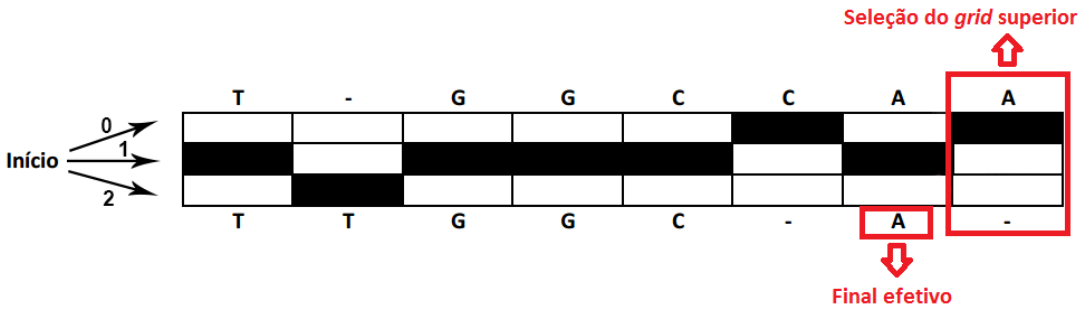


Figura 3.8: Final efetivo na sequência inferior.

as sequências simultaneamente, significa que ela chegou ao fim do caminho.

Quando uma formiga chega ao fim do caminho, um alinhamento entre as duas sequências processadas é obtido. Com isso, o algoritmo pode avançar para a etapa de pontuação do alinhamento par-a-par gerado e atualização da intensidade do feromônio.

3.3.3 Cálculo da Pontuação do Alinhamento par-a-par e Atualização do Feromônio

Uma vez obtido o alinhamento par-a-par das duas sequências analisadas, a pontuação do alinhamento é calculada de modo que as posições de correspondência são pontuadas e as posições de não correspondência ou que contenham *gaps* são penalizadas. Isto pode ser observado na Equação 3.3, em que x representa o resíduo da primeira sequência e y representa o resíduo da segunda sequência.

$$\sigma(x,y) = \begin{cases} 2, & \text{se } x = y \\ -1, & \text{se } x \neq y \\ -1, & \text{se } x = \text{'-'} \cup y = \text{'-'} \end{cases} \tag{3.3}$$

Dado esse esquema de avaliação, o algoritmo calcula a pontuação geral do alinhamento por meio da Equação 3.4, em que S' e T' são a primeira e a segunda sequências alinhadas, respectivamente, e len é o tamanho do alinhamento.

$$Score = \sum_{i=1}^{len} \sigma(S'[i], T'[i]) \tag{3.4}$$

Uma vez que a pontuação do alinhamento de determinado par de sequências é obtida, o algoritmo verifica se a solução encontrada é melhor do que a obtida pela formiga anterior. Se essa condição for verdadeira, o novo alinhamento par-a-par é armazenado em uma estrutura que guarda as informações do melhor alinhamento para um dado par de sequências. Caso a pontuação do novo alinhamento seja pior do que a do alinhamento par-a-par obtido na iteração anterior, o antigo alinhamento continua como a melhor solução.

Quando a pontuação for calculada e o melhor alinhamento armazenado, o algoritmo realiza a atualização da intensidade da trilha de feromônio, com base na Equação 3.5, em que ρ é um parâmetro definido pelo usuário e trata-se de um número em ponto flutuante, entre 0 e 1, que representa o fator de decaimento (evaporação) do feromônio.

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij} \tag{3.5}$$

Nesse contexto, $\Delta\tau_{ij}$ é definido pela Equação 3.6, em que Q e Max_Score são constantes positivas definidas pelo usuário para controle do algoritmo e $Score^z$ é a pontuação obtida pela formiga z, previamente calculada pela Equação 3.4.

$$\Delta\tau_{ij} = \begin{cases} Q * (Score^z + Max\_Score)/2 & \text{se a formiga } z \text{ passar pelo } grid\ (i,j) \\ 0 & \text{se a formiga } z \text{ não passar pelo } grid\ (i,j) \end{cases} \tag{3.6}$$

Uma vez que a trilha de feromônio é atualizada, outra formiga irá iniciar um percurso com base na nova intensidade de feromônio. É importante ressaltar que o número de formigas junto ao algoritmo é definido pelo usuário.

Quando todas as formigas terminarem o trajeto, uma iteração do algoritmo está completa. Dessa forma, o modelo de OCF implementado no presente trabalho repete os mesmos passos na próxima iteração, a partir do processamento das formigas, elucidado na Subseção 3.3.2. É importante ressaltar que o número de iterações do algoritmo também é definido pelo usuário.

Ao término do processo, é armazenado o melhor alinhamento encontrado pelo algoritmo para o par de sequências processado. Desse modo, deve-se repetir todos os passos, a partir daqueles elucidados na Subseção 3.3.1, para todos os possíveis pares das sequências de entrada, de modo a se obter os melhores alinhamentos para todos os pares.

Devido ao fato do algoritmo proposto Zhou et al. (2010) adequar-se apenas a alinhamentos par-a-par, foi utilizado no presente trabalho o algoritmo *Center Star* (ZOU et al., 2015) para a montagem do alinhamento múltiplo das sequências de entrada a partir dos melhores alinhamentos par-a-par obtidos com a OCF.

3.3.4 Alinhamento Múltiplo de Sequências com o Algoritmo *Center Star*

O algoritmo *Center Star* (ZOU et al., 2015) baseia-se na sequência mais similar para realizar a montagem do alinhamento múltiplo. Isso significa que a sequência com a maior pontuação será selecionada como *estrela central* pelo algoritmo e todas as outras sequências do AMS serão compostas pelas sequências advindas dos alinhamentos par-a-par da *estrela central*.

Conforme pode-se observar no exemplo da Figura 3.9, em que para dadas sequências a serem alinhadas, tem-se as pontuações de cada alinhamento par-a-par, e a soma das pontuações de cada sequência. Nesse caso, a sequência S_1 é a sequência de maior somatório de pontuação e, portanto, será definida como *estrela central*.

Uma vez que a sequência S_1 foi definida como *estrela central*, utiliza-se as sequên-

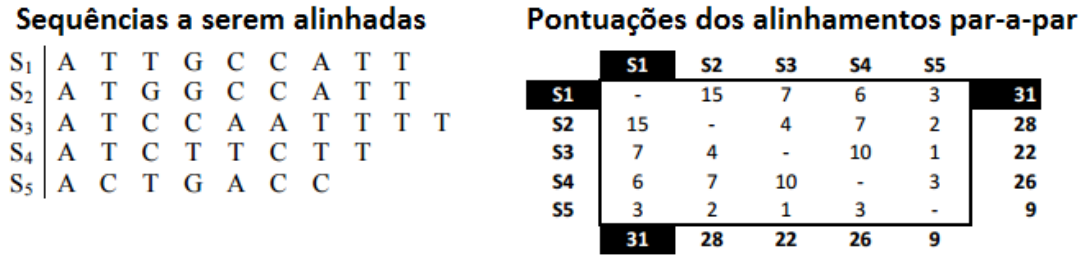


Figura 3.9: Exemplo de sequências a serem alinhadas com o *Center Star*.

cias advindas dos pares de S_1 para se realizar a montagem do alinhamento múltiplo, conforme observa-se na Figura 3.10.

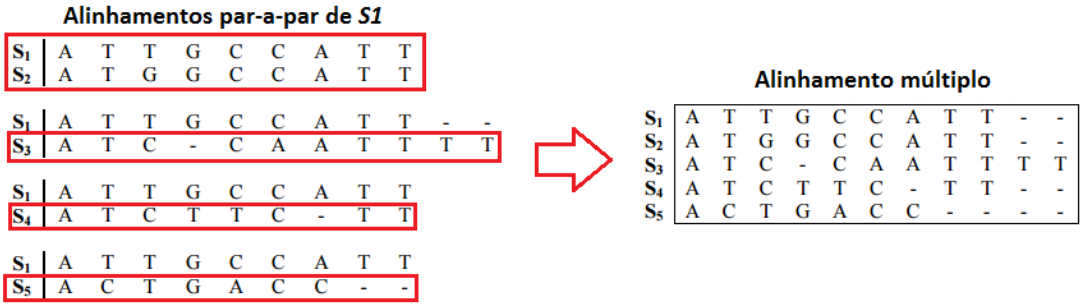


Figura 3.10: Alinhamento múltiplo obtido com as sequências do exemplo.

Assim, é possível se obter alinhamentos múltiplos de sequências a partir das informações de similaridade entre as sequências envolvidas no processamento.

3.4 Considerações Finais

Esse capítulo apresentou o trabalho implementado, no que diz respeito ao escalonador automático de funções objetivo e à etapa de pós processamento com OCF. O escalonador foi desenvolvido com base em um modelo matemático proposto para o cálculo de similaridades. Assim, caso o percentual de similaridade esteja abaixo do limiar indicado pelo usuário por parâmetro, a MSA-GA irá selecionar a função objetivo COFFEE, caso contrário, a função WSP. Com relação à etapa de pós processamento com OCF, os resultados produzidos pelo AG da MSA-GA são dados como entrada para a

OCF, em uma tentativa de se refinar o alinhamento produzido.

Capítulo 4

Testes e Resultados

4.1 Considerações Iniciais

Neste capítulo são apresentados os testes e resultados obtidos a partir da implementação das abordagens propostas no presente trabalho. Além disso, especifica-se a plataforma de testes e o *Benchmark* utilizado para aferir a qualidade dos resultados obtidos.

4.2 Plataforma de Testes

Todos os casos de teste do presente trabalho foram executados no Laboratório de Bioinformática da UNESP, Campus de São José do Rio Preto. A máquina utilizada é um computador Lenovo G40, com sistema operacional Windows 8.1, de arquitetura 64 *bits*, processador Intel Core i5, 4GB de memória RAM.

Na Tabela 4.1 apresenta-se os valores padrão utilizados para cada parâmetro do algoritmo genético da MSA-GA (GONDRO; KINGHORN, 2007). Os termos em Inglês foram mantidos devido ao país de origem em que a ferramenta foi desenvolvida.

O parâmetro *Population Size* representa a quantidade de populações do AG. O parâmetro *Number of Generations* indica o número de iterações que o algoritmo irá executar. Os operadores *Block Mutation*, *Gap Extension Mutation*, *Gap Reduction*

Tabela 4.1: Valores dos parâmetros utilizados no Algoritmo Genético.

Parâmetro	**Valor**
Population Size	1000
Number of Generations	10000
Block Mutation	0,01
Gap Extension Mutation	0,05
Gap Reduction Mutation	0,05
Horizontal Recombination	0,08
Vertical Recombination	0,08
Horizontal/Vertical Ratio	0,05
Tournament Size	2
Offset	0
Maximum Sequence Size	1,2

Mutation, *Horizontal Recombination* e *Vertical Recombination* simbolizam as probabilidades de ocorrência de uma mutação de blocos, de uma mutação por extensão de *gaps*, de uma mutação de redução de *gaps*, de ocorrer uma recombinação horizontal e de ocorrer uma recombinação vertical, respectivamente. O parâmetro *Horizontal/-Vertical Ratio* se refere à proporção de ocorrência dos operadores de recombinação horizontal e vertical. O parâmetro *Tournament Size* indica o número de candidatos que serão selecionados na fase de torneio do AG. O parâmetro *Offset* indica o número máximo de *gaps* inseridos no início das sequências. Por fim, o parâmetro *Maximum Sequence Size* relaciona-se com o tamanho máximo das sequências.

Na Tabela 4.2 observa-se os valores utilizados nos parâmetros da OCF, os quais são os mesmos utilizados por Zhou et al. (2010).

Tabela 4.2: Valores dos parâmetros utilizados na Otimização por Colônia de Formigas.

Parâmetro	**Valor**
Nº de Formigas	20
α	1
β	5
q_0	0,3
ρ	0,2
Q	0,0001
τ_0	10
MAX_SCORE	200

Além do item relativo ao número de formigas do algoritmo, tem-se os parâmetros α e β, relativos ao peso do feromônio, q_0 representa a probabilidade de se utilizar a informação de feromônio, ρ representa o fator de decaimento (evaporação) do feromônio, Q e MAX_SCORE são constantes usadas na atualização do feromônio e τ_0 é a intensidade inicial do feromônio.

4.3 Escalonador de Funções Objetivo

Para validar o funcionamento do escalonador de funções objetivo implementado no presente trabalho, foram utilizados os conjunto de teste do BAliBase (THOMPSON et al., 2005). Esse *Benchmark* oferece diferentes conjuntos de sequências, divididos em categorias de conjuntos mais similares ou menos similares. É importante ressaltar que o BAliBase utiliza um limiar de cerca de 35% para dividir os conjuntos de sequências mais ou menos similares.

Assim, nos testes executados, confrontou-se os resultados obtidos pelo escalonador multifunção com os valores de similaridades dos conjuntos de sequências menos correlatas disponíveis no BAliBase, conforme pode ser verificado na Figura 4.1.

Deste modo, foram executados quinze casos de teste, dos quais oito dos valores obtidos pelo escalonador automático são iguais aos valores apresentados no BAliBase, o que resulta em 53% de exatidão. Além disso, ressalta-se que nos casos em que não houve coincidência de similaridade, a média da divergência foi de 2,85% com um desvio padrão de 1,57, o que implica que mesmo nesses casos os valores analisados se mantêm muito próximos. Os percentuais dos casos em que houve diferença podem ser observados na Tabela 4.3.

Uma explicação cabível para essa divergência é a possível diferença entre valores dos parâmetros utilizados para a realização dos alinhamentos par-a-par, visto que essa informação não se encontra disponível no BAliBase. Ademais, é importante ressaltar que em nenhum caso o valor de similaridade obtido pelo escalonador ultrapassou o

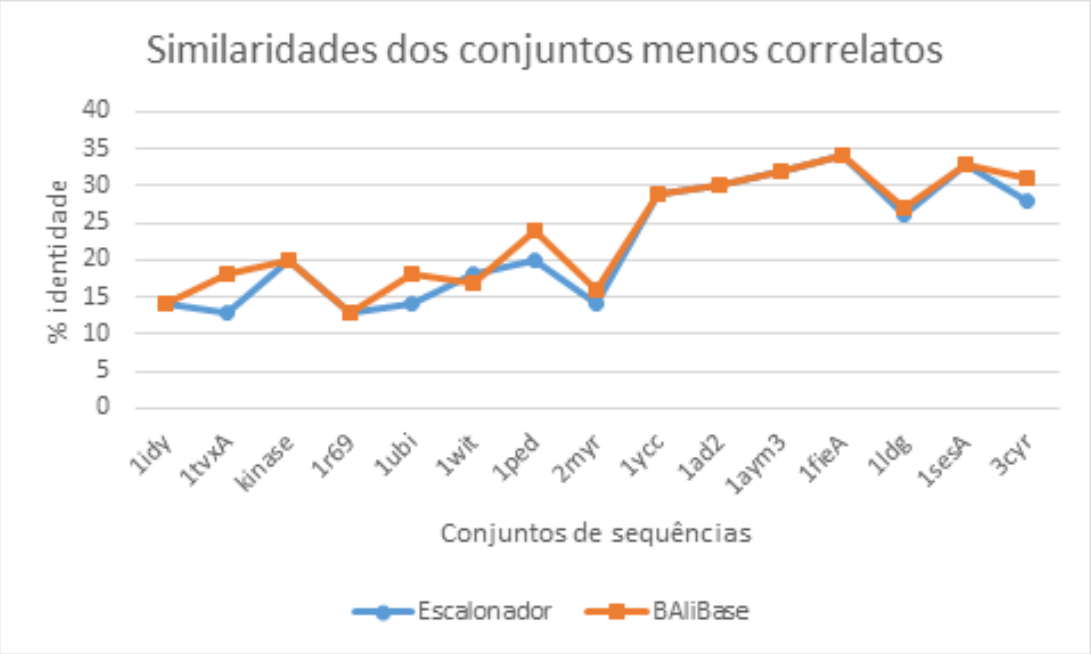


Figura 4.1: Percentuais de identidade dos conjuntos de sequências menos similares.

Tabela 4.3: Percentual de divergência de similaridades entre os conjuntos menos similares.

Conjunto de Sequências	**Percentual da Discrepância (%)**
1tvxA	5
1ubi	4
1wit	1
1ped	4
2myr	2
1ldg	1
3cyr	3

limiar de 35%.

Além dos testes realizados com conjuntos de sequências menos correlatas, foram executados testes para avaliar o comportamento do escalonador de funções junto aos conjuntos de sequências mais similares, conforme observa-se na Figura 4.2.

Nota-se na Figura 4.2 que foram executados quinze casos de teste, dos quais sete dos valores obtidos pelo escalonador automático são iguais aos valores apresentados no BAliBase, o que resulta em 47% de exatidão. Além disso, assim como nos conjuntos menos similares, percebe-se que a discrepância entre os percentuais de identidade

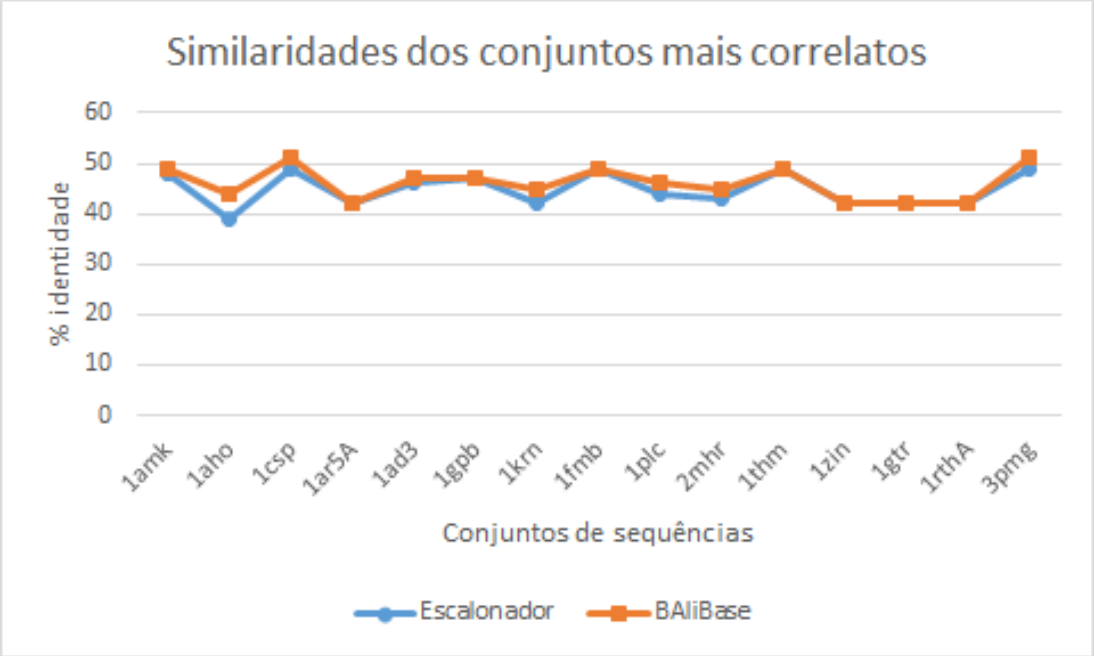


Figura 4.2: Percentuais de identidade dos conjuntos de sequências mais similares.

é em média 2,25% com desvio padrão de 1,28, o que pode ser considerado uma divergência pequena. Os percentuais dos casos em que houve diferença podem ser observados na Tabela 4.4. Ademais, destaca-se também que em nenhum dos casos analisados os valores obtidos ficam abaixo do limiar de 35%.

Portanto, observa-se que o escalonador de funções objetivo, implementado a partir do modelo proposto no presente trabalho, é capaz de calcular a porcentagem de similaridade do conjunto de sequências a ser alinhado, com uma taxa de acerto bastante

Tabela 4.4: Percentual de divergência de similaridades entre os conjuntos mais similares.

Conjunto de Sequências	**Percentual da Discrepância (%)**
1amk	1
1aho	5
1csp	2
1ad3	1
1krn	3
1plc	2
2mhr	2
3pmg	2

considerável. O cálculo correto deste valor é importante para que a melhor função objetivo possa ser selecionada para cada caso específico.

4.4 Refinamento do Alinhamento com Otimização por Colônia de Formigas

Para avaliar as melhorias obtidas com a fase de pós processamento com OCF implementada junto à MSA-GA, foram utilizados os casos de teste do BAliBase.

Foram selecionados, de maneira aleatória, dez casos de teste, com número de sequências e tamanho variados. Para aferir a qualidade dos alinhamentos obtidos foi utilizado no presente trabalho o mesmo modelo de pontuação usado por Zhou et al. (2010), o qual se encontra representado pela Equação 3.3, previamente definida na Subseção 3.3.3. Basicamente, o objetivo é pontuar as posições de correspondências e penalizar as posições em que não há correspondência entre bases ou que apresentem *gaps*. Esse modelo mostrou-se adequado pois destaca a quantidade de correspondências no alinhamento. Devido à abordagem estocástica da OCF, cada caso de teste foi executado cinco vezes e a pontuação considerada foi a média aritmética das execuções, de modo a garantir uma avaliação mais robusta em termos estatísticos.

Com isso, foram comparados os resultados obtidos pela MSA-GA padrão com os resultados obtidos pela ferramenta com o refinamento por OCF, conforme pode ser observado na Tabela 4.5. É importante ressaltar que a melhor pontuação para cada caso de teste está destacada em negrito.

Em nove dos dez casos de teste avaliados, a etapa de pós processamento com OCF foi capaz de refinar o alinhamento produzido pelo AG da MSA-GA, de modo que a melhoria obtida chegou a cerca de 12% no caso de teste *1aym3*. A partir dos resultados obtidos, foram gerados gráficos com as pontuações da MSA-GA padrão e da MSA-GA OCF para cada caso de teste, conforme observa-se na Figura 4.3.

Tabela 4.5: Pontuações obtidas para cada caso de teste.

	MSA-GA	MSA-GA OCF
1ad2	360,0	360,0
1ad3	2133,6	**2266,8**
1aho	114,0	**122,0**
1amk	3358,8	**3365,0**
1ar5A	1302,0	**1304,4**
1aym3	557,4	**624,0**
1csp	640,2	**663,8**
1fieA	2464,2	**2481,0**
1gpb	5206,8	**5238,0**
1ldg	252,0	**258,0**

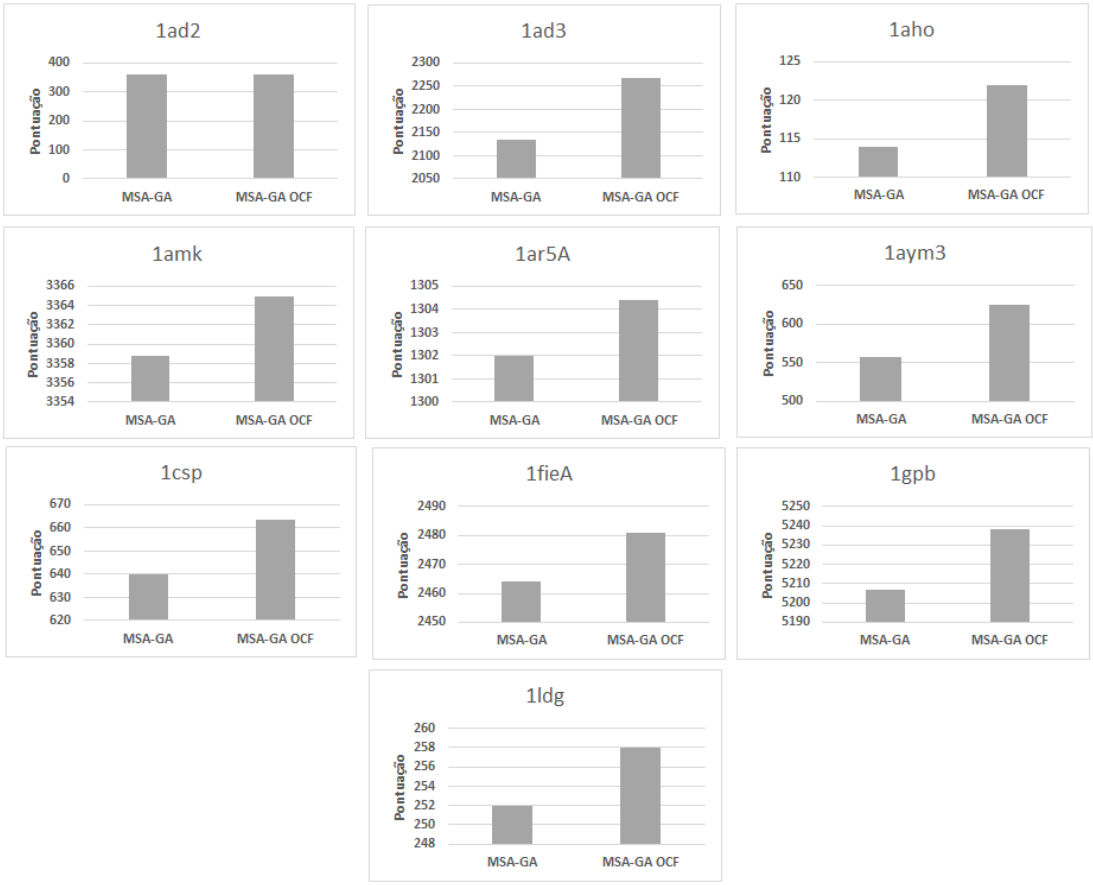


Figura 4.3: Pontuações obtidas pela MSA-GA padrão e pela MSA-GA OCF.

A melhoria observada na pontuação é importante, pois implica na execução de alinhamentos mais corretos. Consequentemente, isto permite que análises mais precisas sejam conduzidas por parte dos biólogos, fato que é de grande importância no

momento de se realizar inferências sobre os dados analisados.

Além disso, foram calculados os desvios padrão de todas as execuções para cada caso de teste, conforme observa-se na Tabela 4.6.

Tabela 4.6: Desvios padrão das execuções de cada caso de teste.

	MSA-GA	**MSA-GA OCF**
1ad2	0,0	0,0
1ad3	33,2	139,2
1aho	0,0	0,0
1amk	22,8	10,1
1ar5A	0,0	5,3
1aym3	89,8	0,0
1csp	4,0	16,9
1fieA	412,7	72,0
1gpb	333,6	0,0
1ldg	0,0	8,7

Percebe-se que a média dos desvios padrão calculados a partir das pontuações obtidas pela MSA-GA em todos os casos de teste foi 89,6, enquanto a média dos desvios padrão da MSA-GA OCF foi 25,2, o que significa que além de produzir alinhamentos com maior significância biológica, a Otimização por Colônia de Formigas implementada no presente trabalho oferece maior robustez à ferramenta, de modo a torná-la mais estável quando comparada com a versão original.

4.5 Considerações Finais

A partir da análise dos resultados, pode-se observar a efetividade das estratégias implementadas no presente trabalho. Por meio do modelo matemático desenvolvido para o cálculo de similaridades das sequências envolvidas no alinhamento múltiplo, possibilitou-se a obtenção dos percentuais de identidade dos conjuntos de sequências de maneira exata ou muito próxima aos valores disponíveis no BAliBase. Com isso, a ferramenta MSA-GA torna-se mais robusta, no sentido de ser capaz de selecionar qual a melhor função objetivo a ser executada para cada alinhamento específico, se-

gundo critérios de similaridade. Além disso, a partir do pós processamento com OCF, possibilitou-se o refinamento dos alinhamentos produzidos pela MSA-GA, o que implica em resultados mais corretos em termos de significância biológica.

Capítulo 5

Conclusão

5.1 Conclusões Gerais

No presente trabalho foram apresentados conceitos fundamentais necessários para o entendimento do escopo da Bioinformática. Além disso, elucidou-se as principais técnicas para alinhamento de sequências e algumas heurísticas vastamente utilizadas para solucionar problemas de alinhamentos de múltiplas sequências, contextualizando-as no seu atual estado-da-arte. Uma atenção especial foi voltada para a heurística de Algoritmos Genéticos, que é parte principal do presente trabalho.

Nesse contexto, a ferramenta MSA-GA destaca-se por utilizar uma abordagem computacionalmente mais simples de AG e ainda assim ser capaz de produzir melhores resultados quando comparada a outras ferramentas muito utilizadas em Bioinformática, como a ClustalW (GONDRO; KINGHORN, 2007). No entanto, Amorim et al. (2015) verificaram que o uso de uma função objetivo mais apropriada para cada alinhamento específico tende a produzir melhores resultados.

Assim, foi implementado no presente trabalho um escalonador automático de funções, junto à MSA-GA, com base no percentual de similaridade das sequências envolvidas, o qual é calculado por meio de um modelo matemático. A partir deste escalonador, possibilita-se que a MSA-GA selecione automaticamente, durante a execução,

a função objetivo mais apropriada para os conjuntos de sequências menos ou mais similares.

Observa-se que o modelo implementado obteve, no geral, percentuais de similaridade idênticos aos dados contidos no BAliBase e, nos casos em que não houve coincidência, os valores permaneceram muito próximos. Além disso, é importante ressaltar que em nenhum dos casos os percentuais de similaridade ultrapassaram o limiar de 35% que divide os casos de teste do BAliBase em conjuntos mais ou menos similares.

Portanto, pode-se afirmar que a principal contribuição em relação ao uso do escalonador automático junto à MSA-GA é oferecer maior robustez à ferramenta, visto que, no geral, permite que a ferramenta selecione automaticamente a melhor função objetivo a ser utilizada para cada alinhamento específico.

Além disso, foi implementada no presente trabalho uma etapa de pós processamento com base em Otimização por Colônia de Formigas, junto à MSA-GA, a fim de se amenizar o problema de máximo local, comumente observado em Algoritmos Genéticos. Observou-se que, em nove dos dez casos de teste executados, a etapa de pós processamento com OCF foi capaz de refinar o alinhamento produzido pela MSA-GA.

Assim, pode-se afirmar que a principal contribuição com relação à etapa de pós processamento com OCF é produzir alinhamentos mais precisos em eventuais casos de máximo local, observados com a MSA-GA. Com isso, é possível se obter maior significância biológica nos resultados, o que implica em análises mais precisas por parte dos biólogos.

5.2 Atividades Futuras

Devido à natureza paralelizável da OCF, pretende-se utilizar estratégias de programação paralela para otimizar o processo também em termos de tempo de execução.

Além disso, pretende-se examinar a eficácia da etapa de pós processamento com OCF junto a outras heurísticas, como Alinhamento Progressivo, entre outras.

Por fim, propõe-se como atividade futura um estudo mais aprofundado na implicação da alteração dos valores dos parâmetros do AG e da OCF para a qualidade do alinhamento produzido.

5.3 Publicações

É importante ressaltar que o presente trabalho contribuiu, em partes, para algumas publicações relacionadas ao período em que o aluno cursou o Mestrado. Um artigo intitulado "*A parallel approach of COFFEE objective function to multiple sequence alignment*" foi publicado na "*International Conferece on Mathematical Modeling in Physical Sciences*", no ano de 2015. Este artigo foi enviado pela própria conferência para um segundo *peer-review* e aceito para publicação na revista "*Journal of Physics: Conference Series*", no mesmo ano. Do mesmo modo, o aluno publicou um artigo intitulado "*Performance Improvement of Genetic Algorithm for Multiple Sequence Alignment*"junto à *International Conference on Parallel and Distributed Computing, Applications and Technologies*, que ocorreu no ano de 2016, na cidade de Guangzhou, na China.

Além disso, o aluno submeteu recentemente um artigo intitulado "*An approach for COFFEE objective function to global DNA multiple sequence alignment*" junto à revista "*Journal of Computational Biology and Chemistry*", o qual encontra-se em fase de avaliação.

Referências Bibliográficas

ADAMS, R. L. P. et al. *The biochemistry of the nucleic acids*. [S.l.]: Chapman and Hall, 1992.

AMORIM, A. et al. Improvements in the sensibility of msa-ga tool using coffee objective function. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2015. v. 574, n. 1, p. 012104.

BAJORATH, J.; STENKAMP, R.; ARUFFO, A. Knowledge-based model building of proteins: concepts and examples. *Protein science: a publication of the Protein Society*, Blackwell Publishing, v. 2, n. 11, p. 1798, 1993.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, ACM, v. 35, n. 3, p. 268–308, 2003.

BOYCE, K.; SIEVERS, F.; HIGGINS, D. G. Instability in progressive multiple sequence alignment algorithms. *Algorithms for Molecular Biology*, BioMed Central Ltd, v. 10, n. 1, p. 26, 2015.

CAVUSLAR, G.; CATAY, B.; APAYDIN, M. S. A tabu search approach for the nmr protein structure-based assignment problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, IEEE Computer Society Press, v. 9, n. 6, p. 1621–1628, 2012.

CHOWDHURY, B.; GARAI, G. A genetic approach with controlled crossover and guided mutation for biological sequence alignment. In: IEEE. *Emerging Applications of Information Technology (EAIT), 2014 Fourth International Conference of*. [S.l.], 2014. p. 307–312.

CORREA, J. M. et al. Parallel simulated annealing for fragment based sequence alignment. In: IEEE. *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*. [S.l.], 2012. p. 641–648.

CRISTINO, A. dos S. *Principais algoritmos de alinhamento de sequências genéticas*. http://www.ime.usp.br/˜alexsc - acessado em 20/01/2012, 2012.

DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. *Theoretical computer science*, Elsevier, v. 344, n. 2, p. 243–278, 2005.

DORIGO, M.; GAMBARDELLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 1, n. 1, p. 53–66, 1997.

EDGAR, R. C.; BATZOGLOU, S. Multiple sequence alignment. *Current opinion in structural biology*, Elsevier, v. 16, n. 3, p. 368–373, 2006.

EIDHAMMER, I.; JONASSEN, I.; TAYLOR, W. R. Structure comparison and structure patterns. *Journal of Computational Biology*, Mary Ann Liebert, Inc., v. 7, n. 5, p. 685–716, 2000.

FENG, D.-F.; DOOLITTLE, R. F. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, Springer, v. 25, n. 4, p. 351–360, 1987.

FINK, G. A. *Markov models for pattern recognition: from theory to applications*. [S.l.]: Springer Science & Business Media, 2014.

GHOUZAM, Y. et al. Improving protein fold recognition with hybrid profiles combining sequence and structure evolution. *Bioinformatics*, Oxford Univ Press, p. btv462, 2015.

GLENN, T. C. Field guide to next-generation dna sequencers. *Molecular ecology resources*, Wiley Online Library, v. 11, n. 5, p. 759–769, 2011.

GLOVER, F.; LAGUNA, M. *Tabu search*. [S.l.]: Springer, 1999.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. *Machine learning*, Springer, v. 3, n. 2, p. 95–99, 1988.

GONDRO, C.; KINGHORN, B. A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research*, v. 6, n. 4, p. 964–982, 2007.

GOULD, S. J. *Ontogeny and phylogeny*. [S.l.]: Harvard University Press, 1977.

GUSFIELD, D. *Algorithms on strings, trees and sequences: computer science and computational biology*. [S.l.]: Cambridge university press, 1997.

HOSEINI, P.; SHAYESTEH, M. G. Efficient contrast enhancement of images using hybrid ant colony optimisation, genetic algorithm, and simulated annealing. *Digital Signal Processing*, Elsevier, v. 23, n. 3, p. 879–893, 2013.

ISHIKAWA, M. et al. Multiple sequence alignment by parallel simulated annealing. *Computer applications in the biosciences: CABIOS*, Oxford Univ Press, v. 9, n. 3, p. 267–273, 1993.

JANGAM, S.; CHAKRABORTI, N. A novel method for alignment of two nucleic acid sequences using ant colony optimization and genetic algorithms. *Applied Soft Computing*, Elsevier, v. 7, n. 3, p. 1121–1130, 2007.

KALOS, M. H. Monte carlo methods in the physical sciences. In: IEEE PRESS. *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*. [S.l.], 2007. p. 266–271.

KATOH, K. et al. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, Oxford Univ Press, v. 30, n. 14, p. 3059–3066, 2002.

KATOH, K.; STANDLEY, D. M. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, SMBE, v. 30, n. 4, p. 772–780, 2013.

KATOH, K.; STANDLEY, D. M. Mafft: iterative refinement and additional methods. In: *Multiple Sequence Alignment Methods*. [S.l.]: Springer, 2014. p. 131–146.

KATOH, K.; STANDLEY, D. M. A simple method to control over-alignment in the mafft multiple sequence alignment program. *Bioinformatics*, Oxford Univ Press, p. btw108, 2016.

KATOH, K.; TOH, H. Parallelization of the mafft multiple sequence alignment program. *Bioinformatics*, Oxford Univ Press, v. 26, n. 15, p. 1899–1900, 2010.

KAYA, M.; KAYA, B.; ALHAJJ, R. A novel multi-objective genetic algorithm for multiple sequence alignment. *International Journal of Data Mining and Bioinformatics*, Inderscience Publishers (IEL), v. 14, n. 2, p. 139–158, 2016.

KAYA, M.; SARHAN, A.; ALHAJJ, R. Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. *Computer methods and programs in biomedicine*, Elsevier, v. 114, n. 1, p. 38–49, 2014.

KELLER, O. et al. A novel hybrid gene prediction method employing protein multiple sequence alignments. *Bioinformatics*, Oxford Univ Press, v. 27, n. 6, p. 757–763, 2011.

KHURI, S. A bioinformatics track in computer science. In: ACM. *ACM SIGCSE Bulletin*. [S.l.], 2008. v. 40, n. 1, p. 508–512.

KIM, J.; PRAMANIK, S.; CHUNG, M. J. Multiple sequence alignment using simulated annealing. *Computer applications in the biosciences: CABIOS*, Oxford Univ Press, v. 10, n. 4, p. 419–426, 1994.

KIRKPATRICK, S. et al. Optimization by simulated annealing. *science*, World Scientific, v. 220, n. 4598, p. 671–680, 1983.

LEE, A.; KING, L. Aco implementation for sequence alignment with genetic algorithms. *arXiv preprint arXiv:1406.0930*, 2014.

LEE, Z.-J. et al. Genetic algorithm with ant colony optimization (ga-aco) for multiple sequence alignment. *Applied Soft Computing*, Elsevier, v. 8, n. 1, p. 55–78, 2008.

LEMOS, M.; ARAGAO, M. V. S. P.; CASANOVA, M. A. *Padrões em Biossequências*. [S.l.]: PUC, 2003.

LEMOS, M.; BASÍLIO, A.; CASANOVA, M. A. *Um estudo dos algoritmos de montagen de fragmentos de DNA*. [S.l.]: PUC, 2003.

LEMOS, M.; CASANOVA, M. A. *Algoritmos para análise de sequências*. [S.l.]: PUC, 2000.

LIEW, A. W.-C.; YAN, H.; YANG, M. Pattern recognition techniques for the emerging field of bioinformatics: A review. *Pattern Recognition*, Elsevier, v. 38, n. 11, p. 2055–2073, 2005.

LIN, X.; ZHANG, X. et al. Protein structure prediction with local adjust tabu search algorithm. *BMC bioinformatics*, BioMed Central Ltd, v. 15, n. Suppl 15, p. S1, 2014.

LIU, Y.; POPP, B.; SCHMIDT, B. Cushaw3: sensitive and accurate base-space and color-space short-read alignment with hybrid seeding. 2014.

LV, Q. et al. A parallel ant colonies approach to de novo prediction of protein backbone in casp8/9. *Science China Information Sciences*, Springer, v. 56, n. 10, p. 1–13, 2013.

MAREUIL, F. et al. Grid computing for improving conformational sampling in nmr structure calculation. *Bioinformatics*, Oxford Univ Press, v. 27, n. 12, p. 1713–1714, 2011.

MARUCCI, E. et al. Routine libraries for pattern recognition in quasispecies. *Genetics and molecular research: GMR*, p. 970–981, 2008.

MORGENSTERN, B. et al. Dialign: finding local similarities by multiple sequence alignment. *Bioinformatics*, Oxford Univ Press, v. 14, n. 3, p. 290–294, 1998.

MORRISON, D.; MORGAN, M.; KELCHNER, S. Molecular homology and multiple sequence alignment: an analysis of concepts and practice. *Australian Systematic Botany*, CSIRO, 2015.

MOSS, J.; JOHNSON, C. G. An ant colony algorithm for multiple sequence alignment in bioinformatics. In: SPRINGER. *Artificial Neural Nets and Genetic Algorithms*. [S.l.], 2003. p. 182–186.

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*, Elsevier, v. 48, n. 3, p. 443–453, 1970.

NOTREDAME, C.; HIGGINS, D. G. Saga: sequence alignment by genetic algorithm. *Nucleic acids research*, Oxford Univ Press, v. 24, n. 8, p. 1515–1524, 1996.

NOTREDAME, C.; HIGGINS, D. G.; HERINGA, J. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, Elsevier, v. 302, n. 1, p. 205–217, 2000.

NOTREDAME, C.; HOLM, L.; HIGGINS, D. G. Coffee: an objective function for multiple sequence alignments. *Bioinformatics*, Oxford Univ Press, v. 14, n. 5, p. 407–422, 1998.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. *Combinatorial optimization: algorithms and complexity*. [S.l.]: Courier Corporation, 1998.

PIERRI, C. L.; PARISI, G.; PORCELLI, V. Computational approaches for protein function prediction: a combined strategy from multiple sequence alignment to molecular docking-based virtual screening. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, Elsevier, v. 1804, n. 9, p. 1695–1712, 2010.

PROSDOCIMI, F. et al. Bioinformática: manual do usuário. *Biotecnologia Ciência & Desenvolvimento*, v. 29, p. 12–25, 2002.

RIAZ, T.; WANG, Y.; LI, K.-B. Multiple sequence alignment using tabu search. In: AUSTRALIAN COMPUTER SOCIETY, INC. *Proceedings of the second conference on Asia-Pacific bioinformatics-Volume 29*. [S.l.], 2004. p. 223–232.

RIAZ, T.; YI, W.; LI, K.-B. A tabu search algorithm for post-processing multiple sequence alignment. *Journal of bioinformatics and computational biology*, World Scientific, v. 3, n. 01, p. 145–156, 2005.

RUBIN, G. M. et al. Comparative genomics of the eukaryotes. *Science*, American Association for the Advancement of Science, v. 287, n. 5461, p. 2204–2215, 2000.

RUBIO-LARGO, Á.; VEGA-RODRÍGUEZ, M. A.; GONZÁLEZ-ÁLVAREZ, D. L. Hybrid multiobjective artificial bee colony for multiple sequence alignment. *Applied Soft Computing*, Elsevier, 2016.

SAIKATKAR, S.; SAHA, S.; CHAKRABARTI, T. Dna multiple sequence alignment by an ant colony based approach. *Biometrics and Bioinformatics*, v. 6, n. 4, p. 132–136, 2014.

SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, SMBE, v. 4, n. 4, p. 406–425, 1987.

SCHMOLLINGER, M. et al. Dialign p: fast pair-wise and multiple sequence alignment using parallel processors. *BMC bioinformatics*, BioMed Central Ltd, v. 5, n. 1, p. 128, 2004.

SCHWARTZ, A. S.; PACHTER, L. Multiple alignment by sequence annealing. *Bioinformatics*, Oxford Univ Press, v. 23, n. 2, p. e24–e29, 2007.

SCHWEFEL, H.-P.; RUDOLPH, G. *Contemporary evolution strategies*. [S.l.]: Springer, 1995.

SHEN, Q.; SHI, W.-M.; KONG, W. Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Computational Biology and Chemistry*, Elsevier, v. 32, n. 1, p. 53–60, 2008.

SIEVERS, F.; HIGGINS, D. G. Clustal omega, accurate alignment of very large numbers of sequences. In: *Multiple sequence alignment methods*. [S.l.]: Springer, 2014. p. 105–116.

SIEVERS, F. et al. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, EMBO Press, v. 7, n. 1, p. 539, 2011.

SIMONS, K. T. et al. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of molecular biology*, Elsevier, v. 268, n. 1, p. 209–225, 1997.

SMITH, T. F.; WATERMAN, M. S. Identification of common molecular subsequences. *Journal of molecular biology*, Elsevier, v. 147, n. 1, p. 195–197, 1981.

SUBRAMANIAN, A. R. et al. Dialign-tx: greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms Mol Biol*, v. 3, n. 6, p. 1–11, 2008.

SUN, J. et al. Multiple sequence alignment using the hidden markov model trained by an improved quantum-behaved particle swarm optimization. *Information Sciences*, Elsevier, v. 182, n. 1, p. 93–114, 2012.

THOMPSON, J. D.; HIGGINS, D. G.; GIBSON, T. J. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, Oxford Univ Press, v. 22, n. 22, p. 4673–4680, 1994.

THOMPSON, J. D. et al. Balibase 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, Wiley Online Library, v. 61, n. 1, p. 127–136, 2005.

THOMSEN, R.; BOOMSMA, W. Multiple sequence alignment using saga: investigating the effects of operator scheduling, population seeding, and crossover operators. In: *Applications of Evolutionary Computing*. [S.l.]: Springer, 2004. p. 113–122.

VASANT, P. M.; GANESAN, T.; ELAMVAZUTHI, I. Hybrid tabu search hopfield recurrent ann fuzzy technique to the production planning problems: a case study of crude oil in refinery industry. *International Journal of Manufacturing, Materials, and Mechanical Engineering (IJMMME)*, IGI Global, v. 2, n. 1, p. 47–65, 2012.

VÊNCIO, R. Z.; OLIVEIRA, G. B. de. Integração entre biologia e física: construindo redes de regulação gênica usando equações diferenciais. *Revista de Ensino de Bioquímica*, v. 4, n. 1, p. 30–43, 2006.

WALLACE, I. M.; BLACKSHIELDS, G.; HIGGINS, D. G. Multiple sequence alignments. *Current opinion in structural biology*, Elsevier, v. 15, n. 3, p. 261–266, 2005.

WANG, L.; JIANG, T. On the complexity of multiple sequence alignment. *Journal of computational biology*, v. 1, n. 4, p. 337–348, 1994.

WATSON, J. D.; CRICK, F. H. et al. Molecular structure of nucleic acids. *Nature*, v. 171, n. 4356, p. 737–738, 1953.

WHITMAN, W. B.; COLEMAN, D. C.; WIEBE, W. J. Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 95, n. 12, p. 6578–6583, 1998.

ZAFALON, G. et al. A parallel approach of coffee objective function to multiple sequence alignment. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2015. v. 633, n. 1, p. 012084.

ZAFALON, G. F. D. Algoritmos de alinhamento múltiplo e técnicas de otimização para esses algoritmos utilizando ant colony. Universidade Estadual Paulista (UNESP), 2009.

ZAFALON, G. F. D. *Aplicação de estratégias híbridas em algoritmos de alinhamento múltiplo de sequências para ambientes de computação paralela e distribuída.* Tese (Doutorado) — Universidade de São Paulo, 2014.

ZHOU, D. et al. Separation of near full-length hepatitis c virus quasispecies variants from a complex population. *Journal of virological methods*, Elsevier, v. 141, n. 2, p. 220–224, 2007.

ZHOU, X. et al. An ant colony pairwise alignment based on the simplified grid. *Journal of Computational and Theoretical Nanoscience*, American Scientific Publishers, v. 7, n. 1, p. 277–280, 2010.

ZHU, H.; HE, Z.; JIA, Y. A novel approach to multiple sequence alignment using multi-objective evolutionary algorithm based on decomposition. IEEE, 2015.

ZHU, H. et al. Paralleling genetic annealing algorithm on grid. In: IEEE. *Intelligent Networks and Intelligent Systems (ICINIS), 2011 4th International Conference on*. [S.l.], 2011. p. 89–92.

ZHU, X. et al. Parallel implementation of mafft on cuda-enabled graphics hardware. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, IEEE, v. 12, n. 1, p. 205–218, 2015.

ZOU, Q. et al. Halign: Fast multiple similar dna/rna sequence alignment based on the centre star strategy. *Bioinformatics*, Oxford Univ Press, p. btv177, 2015.