

UNIVERSIDADE FEEVALE

SAMUEL ANTÔNIO KLEIN

FERRAMENTA DE APOIO NA APLICAÇÃO DE ALGORITMOS DE ALINHAMENTO  
EM BIOLOGIA COMPUTACIONAL

Novo Hamburgo  
2014

SAMUEL ANTÔNIO KLEIN

FERRAMENTA DE APOIO NA APLICAÇÃO DE ALGORITMOS DE ALINHAMENTO  
EM BIOLOGIA COMPUTACIONAL

Trabalho de Conclusão de Curso  
apresentado como requisito parcial  
à obtenção do grau de Bacharel em  
Ciência da Computação pela  
Universidade Feevale.

Orientador: Ricardo Ferreira de Oliveira

Novo Hamburgo  
2014

## **AGRADECIMENTOS**

*Agradeço a todos aqueles que, de alguma forma, estiveram presentes durante a realização deste trabalho.*

*Aos meus familiares, principalmente à minha esposa, pelo incentivo.*

*Ao professor Ms. Ricardo Ferreira de Oliveira, pela orientação e ensinamentos recebidos, tornando possível a realização desta monografia.*

*Aos meus amigos, pelo apoio.*

## RESUMO

A bioinformática é um campo da biologia relacionada à computação que está experimentando um rápido crescimento e desenvolvimento, devido à necessidade de se manipular com grande quantidade de dados genéticos e bioquímicos. A bioinformática abrange um número amplo de tópicos, sendo, um dos principais, o alinhamento de sequências. Alunos da área da Biologia, com pouca experiência em informática, têm razoável dificuldade no aprendizado de algoritmos de alinhamento. Para aperfeiçoar a aprendizagem dos alunos nas disciplinas que envolvem ferramentas de bioinformática, foi criado um ambiente educativo voltado para ilustrar as etapas e processos do alinhamento de cromossomos e proteínas. Com esse ambiente, visa-se proporcionar aos estudantes de bioinformática uma ferramenta. Este trabalho apresentará os métodos de alinhamento utilizados em disciplinas de bioinformática, culminando com o desenvolvimento de uma ferramenta educativa que proporcione uma clara compreensão de como estes algoritmos trabalham para o aluno com pouca formação na área de computação.

**Palavras-chave:** Bioinformática. Biologia Computacional. Apoio ao ensino de Bioinformática. Biologia Molecular. Alinhamento de Sequências.

## ABSTRACT

Bioinformatics is a field of biology related to computing that is experiencing rapid growth and development due to the need to be handled with great amount of genetic and biochemical data. Bioinformatics covers a wide range of topics, the main one being the sequence alignment. Undergraduate students in biology, with little computer experience, face reasonable difficulty in the alignment algorithms learning. To improve student learning in the disciplines involving bioinformatics tools this work intended to create a learning environment aimed to illustrate the steps and processes of alignment of chromosomes and proteins. With this environment, the goal is to give students a tool for Bioinformatics. This work will present the alignment methods used in bioinformatics disciplines culminating in the development of an educational tool that provides a clear understanding of how these algorithms work for the student with little training in computing.

**Key words:** Bioinformatics. Computational Biology. Support teaching of Bioinformatics. Molecular Biology. Alignment of Sequences.

## LISTA DE FIGURAS

Figura 1 - Estrutura química do DNA .....	13
Figura 2 - Estrutura quaternária da Hemoglobina .....	16
Figura 3 - Esquema do Dogma Central da Biologia Molecular .....	17
Figura 4 - Termociclador, Aparelho utilizado para realizar uma PCR.....	18
Figura 5 - Separação por densidade de moléculas através da eletroforese em gel.....	19
Figura 6 - Comparação entre alinhamento local e alinhamento global.....	24
Figura 7 - Alinhamento local, com lacunas .....	26
Figura 8 - Alinhamento local, com lacunas .....	26
Figura 9 - Encadeamento no Alinhamento Rápido Global de Sequências.....	27
Figura 10 - Fases recursivas no alinhamento de seqüências eficiente em termos de espaço ...	27
Figura 11 - Diagrama de Fluxo de Dados.....	29
Figura 12 - Modelo ER.....	30
Figura 13 - Caso de Uso .....	31
Figura 14 - Diagrama de Classe .....	31
Figura 15 - Interface inicial .....	38
Figura 16 – Protótipo de cadastro de novo alinhamento .....	39
Figura 17 - Cadastro de novo alinhamento.....	40
Figura 18 - Detalhe de um nodo selecionado destacando Gap.....	41
Figura 19 - Detalhe de um nodo selecionado destacando Mismatch .....	42
Figura 20 – Protótipo de exemplo de alinhamento global.....	43
Figura 21 - Exemplo de alinhamento global .....	44
Figura 22 – Protótipo detalhe de um nodo selecionado.....	44
Figura 23 - Detalhe de um nodo selecionado .....	45
Figura 24 - Exemplo de alinhamento manual.....	46
Figura 25 – Alinhamento manual (passo 1).....	47
Figura 26 - Alinhamento manual (passo 2) .....	48
Figura 27 - Alinhamento manual (passo 3) .....	48
Figura 28 - Alinhamento manual (passo 4) .....	49
Figura 29 - Alinhamento manual detalhanto (passo 4).....	50
Figura 30 - Exemplo de comparação .....	52
Figura 31 - Disposição das sequencias na matriz .....	52
Figura 32 - Alinhamento global.....	53
Figura 33 - Alinhamento local.....	54
Figura 34 - Exemplo de alinhamento concluído.....	55
Figura 35 - Alinhamento semi-global.....	56
Figura 36 – Distribuição .....	57

## LISTA DE QUADROS

Quadro 1 - Matriz para comparação entre as sequências AGGTG e GCTTG. ....	23
Quadro 2 - Matriz para comparação entre as sequências AGGTG e GCTTG após o primeiro preenchimento .....	23
Quadro 3 - Matriz para comparação entre as sequências AGGTG e GCTTG após o segundo preenchimento .....	24

## LISTA DE ABREVIATURAS E SIGLAS

BLAST - *Basic Local Alignment Search Tool*

CVS - *Concurrent Version System*

DNA - *Ácido Desoxirribonucleico*

GWT - *Google webtoolkit*

JS - *JavaScript*

JSON - *JavaScriptObjectNotation*

PDB - *Protein Data Bank*

RNA - *Ácido Ribonucleico*

SVN - *Subversion*

UML - *Unified Modeling Language*

# SUMÁRIO

<b>INTRODUÇÃO</b>	<b>11</b>
<b>1 CONCEITOS BÁSICOS DE BIOLOGIA MOLECULAR</b>	<b>12</b>
1.1 DNA - ÁCIDO DESOXIRRIBONUCLEICO	12
1.2 RNA- ÁCIDO RIBONUCLEICO	14
1.3 PROTEÍNAS	15
1.3.1 Estrutura primária	15
1.3.2 Estrutura secundária	15
1.3.3 Estrutura terciária	16
1.3.4 Estrutura quaternária	16
1.4 O DOGMA CENTRAL DA BIOLOGIA MOLECULAR	17
1.5 TÉCNICAS DE ANÁLISE DE DNA	18
<b>2 ALINHAMENTO DE SEQUÊNCIAS</b>	<b>21</b>
2.1 BIOINFORMÁTICA	21
2.2 COMPARAÇÃO DE PARES DE SEQUÊNCIAS	21
2.2.1 Comparação global	22
2.2.2 Comparação local	24
2.2.3 Comparação semi-global	25
2.2.4 Outras comparações	26
2.2.4.1 Alinhamento rápido global de sequências	26
2.2.4.2 Alinhamento de sequências eficiente em termos de espaço	27
2.2.4.3 Alinhamento emendado	28
2.2.4.4 Alinhamento segmentar	28
2.2.4.5 Alinhamento em ordem parcial	28
<b>3 METODOLOGIA DA IMPLEMENTAÇÃO</b>	<b>29</b>
3.1 MODELO DO DIAGRAMA DE FLUXO DE DADOS	29
3.2 DIAGRAMA DE ENTIDADE-RELACIONAMENTO	30
3.3 UML- UNIFIED MODELING LANGUAGE	30
3.3.1 Caso de Uso	30
3.3.2 Diagrama de Classe	31
3.4 POLÍTICA OPEN SOURCE	32
3.4.1 Fontes Colaborativos	32
3.4.1.1 SourceForge	32
3.4.1.2 Github	32
3.5 CONTROLE DE VERSÃO	33
3.6 TESTE UNITÁRIO	33
3.7 FERRAMENTA DE AUTOMAÇÃO E GERENCIAMENTO DE DEPENDÊNCIAS	33
3.8 NODE.JS	34
3.9 WEB E NODEWEBKIT	34
3.10 GWT - JAVA	35
3.11 HTML5/CSS3 - JQUERY	35
3.12 SVG -SCALABLE VECTOR GRAPHICS	35
3.13 DADOS ARMAZENADOS	36
3.14 INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	36
3.15 BOOTSTRAP	36
3.16 HANDLEBARS	37
<b>4 A FERRAMENTA DE ALINHAMENTO</b>	<b>38</b>
4.1 INTERFACE DNA-SEQUENCE	38
4.1.1 Novo alinhamento	39
4.1.1.1 Nucleotídeos ou Proteínas	40
4.1.1.2 Sequência A e B	40
4.1.1.3 Método (Global, Local, Semi-global)	41
4.1.1.4 Gap Penalty	41
4.1.1.5 Match e Mismatch	41

4.1.1.6 Exceção Mismatch	42
4.1.2 Deletar alinhamento	42
4.2 ABRIR UM ALINHAMENTO	42
4.2.1 Nodo diagonal	45
4.2.2 Nodo superior e esquerdo	45
4.2.3 Nodo principal	46
4.2.3 Construção do alinhamento manual	47
<b>5 ESTRUTURAS DE ALINHAMENTO DA DNA-SEQUENCE</b>	<b>51</b>
5.1 CÁLCULO DO GRÁFICO DE PONTOS	51
5.2 CÁLCULO DO ALINHAMENTO	52
5.2.1 Cálculo do alinhamento global	53
5.2.2 Cálculo do alinhamento local	54
5.2.3 Cálculo do alinhamento semi-global	56
<b>6 DISTRIBUIÇÃO DO SOFTWARE</b>	<b>57</b>
<b>CONCLUSÃO</b>	<b>58</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>61</b>
<b>APÊNDICE A – ESTRUTURA JSON DO ALINHAMENTO GLOBAL DAS SEQUENCIAS (“ABCDE”, “ABCEF”)</b>	<b>65</b>
<b>APÊNDICE B – ESTRUTURA DAS CLASSES DA FERRAMENTA DNA-SEQUENCE</b>	<b>67</b>
<b>APÊNDICE C – ARQUIVO DE PARÂMETRO INTERNACIONALIZAÇÃO</b>	<b>70</b>
<b>APÊNDICE D – AVALIAÇÃO DA FERRAMENTA DE APOIO AO ENSINO DE TIPOS DE ALINHAMENTO EM BIOINFORMÁTICA</b>	<b>72</b>

## INTRODUÇÃO

A biologia molecular é uma área em crescente expansão e, por isso, necessita cada vez mais de ferramentas que ilustrem melhor alguns dos processos, métodos e algoritmos relacionados a ela. Isso está relacionado ao fato do crescente estudo e o conhecimento dos processos de forma detalhada desta área e tornarem-se uma necessidade na elaboração de modelos e conceitos para o estudo da biologia molecular.

Diante dessa realidade, os biólogos buscam cada vez mais soluções que atendam e auxiliem essa necessidade, através de um mapeamento dos “problemas” que são comuns dentro do processo na biologia. Uma prática comum é a utilização de ferramentas de busca para alinhamentos iguais ou semelhantes em bases de dados, como o *BLAST*. Para verificar os resultados, há uma dificuldade em compreender de que forma estes são identificados e como são construídos estes alinhamentos.

Com base nisso, este trabalho tem como objetivo contribuir para a exemplificação e ilustração dos principais algoritmos de sequenciamento e alinhamento utilizados na área de biologia molecular, aprimorando o ferramental didático sobre o tema.

Dessa forma, foi construído uma ferramenta voltada tanto ao ambiente *web* quanto ao ambiente *desktop*, com o intuito de auxiliar e melhorar o entendimento dos processos de alinhamento. Almeja-se que a ferramenta tenha interfaces amigáveis e seja baseada em conceitos apresentados em livros de bioinformática focados no ensino dos algoritmos de alinhamento de sequências.

A ferramenta apresenta uma estrutura simples com um modelo funcional para um melhor entendimento do processo de alinhamento, com tela de cadastro de alinhamentos apresentando o maior número de informações para a obtenção dos resultados, uma lista com todos os alinhamentos cadastrados e uma interface com a matriz do alinhamento aberto com os resultados.

Um requisito importante desta ferramenta proposta é a independência de instalação de *software* ou *plugin* adicionado ao sistema para a sua utilização. Para tal, basta ter acesso à internet com um *browser* atualizado, obtendo-se, desse modo, o resultado desejado.

## 1 CONCEITOS BÁSICOS DE BIOLOGIA MOLECULAR

O objetivo deste capítulo é apresentar os conceitos de biologia molecular necessários para o entendimento dos problemas relacionados ao alinhamento de sequências. Pretende-se, aqui, ilustrar as principais estruturas estudadas na biologia molecular, como o DNA, o RNA e as proteínas. Será apresentado o dogma central da biologia molecular, assim como tipos de mutações e polimorfismos.

O intuito dessa revisão bibliográfica é proporcionar uma visão geral dos conceitos de biologia molecular, de maneira que gere uma compreensão adequada às questões e problemas que envolvem o alinhamento.

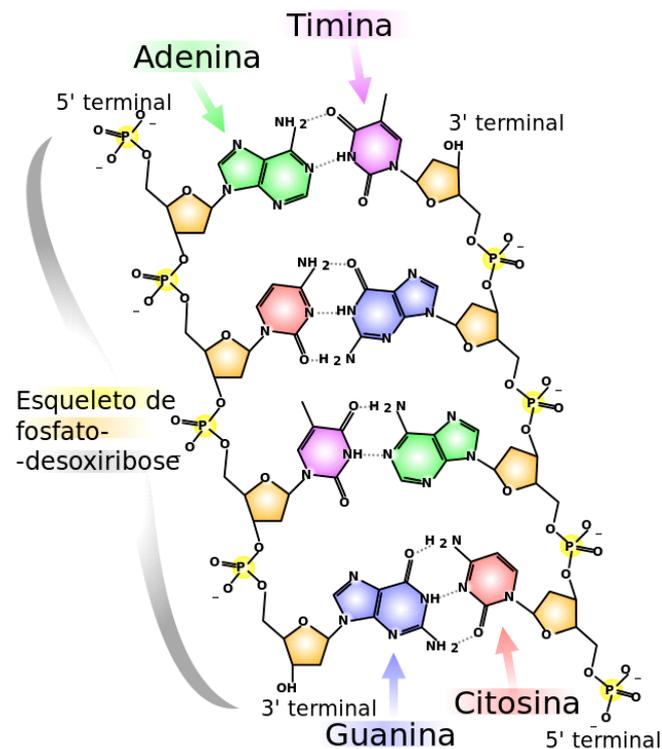
### 1.1 DNA - ÁCIDO DESOXIRRIBONUCLEICO

O DNA, ácido desoxirribonucleico, é uma cadeia de polímeros composta por unidades individuais denominadas nucleotídeos. O código de DNA possui em sua composição quatro tipos de nucleotídeos, denominados: A (adenina), T (timina), G (guanina) e C (citosina).

A molécula deste ácido é disposta em duas cadeias de nucleotídeos, em uma disposição helicoidal. Os nucleotídeos, em cada uma das cadeias, são ligados por ligações 3' e 5' *diéster-fosfato*, sendo que uma cadeia é inversa em relação à outra, denominadas antiparalelas. Assim, um nucleotídeo em uma cadeia pareia com outro na cadeia antiparalela ligada por pontes de hidrogênio. A adenina sempre pareia com a timina, através de duas ligações de hidrogênio e, a guanina, sempre com a citosina, mas com três ligações de hidrogênio, de forma que a última é mais forte que a primeira (JUNQUEIRA, 2011).

A estrutura química da molécula de DNA pode ser vista na Figura 1. É possível observar as três ligações de hidrogênio entre as bases guanina-citosina e o número menor de ligações de hidrogênio (duas) entre as bases adenina-timina.

Figura 1 - Estrutura química do DNA



Fonte: WikimediaCommons

Segundo Junqueira (2011), o DNA é o repositório da informação genética, sendo igualmente responsável pela transmissão da informação genética. Por ser o encarregado da informação da vida, alguns pesquisadores estabelecem um paralelo entre o DNA e o *software* computacional, como Chaitin (2005, p.67):

E como é este software? Ele não está escrito em 0/1 binário como um software de computador. Em vez disso, o DNA está escrito em um alfabeto de 4 letras, as quatro bases que podem ser torcidas em uma escada de dupla-hélice que é uma molécula de DNA [...] E este software é altamente conservador, muito do qual é bastante antigo: muitas sub-rotinas comuns são compartilhadas entre as moscas de fruta, invertebrados, ratos e seres humanos.<sup>1</sup>

Conforme Chandar (2011), a informação genética dos seres vivos é conhecida coletivamente como *genoma*. O genoma das diversas espécies de seres vivos é enorme. Mesmo para seres unicelulares, o volume de dados é impressionante. Para compreensão do comportamento do sistema é necessária informação quantitativa detalhada. De acordo com Alberts (2008), para compreendermos as células necessitamos da matemática, da computação e da informação quantitativa.

<sup>1</sup> Tradução do autor.

## 1.2 RNA- ÁCIDO RIBONUCLEICO

O RNA, ácido ribonucleico, de forma semelhante ao DNA, é uma cadeia de polímeros composta por unidades individuais, denominadas nucleotídeos. O código do RNA, contudo, não tem exatamente as mesmas bases que o DNA, pois no lugar da timina (T) o RNA tem o nucleotídeo uridina (U) (RINGO, 2005).

De forma diversa do DNA, a molécula do RNA é um filamento único, na grande maioria das vezes. Apenas excepcionalmente é que se encontra moléculas de RNA formando filamentos duplos complementares, como é o caso de alguns vírus (JUNQUEIRA, 2011).

Quanto ao ponto de vista funcional e estrutural os principais tipos de RNA são:

O RNA de transferência, ou tRNA, é relativamente menor que os outros dois, comparativamente. É constituído de 75 a 90 nucleotídeos. Tem como função transferir aminoácidos para as posições corretas nas cadeias de proteínas em formação (JUNQUEIRA, 2011).

O RNA mensageiro, ou mRNA, é sintetizado pelos cromossomos no núcleo celular e contém a transcrição de um segmento de uma das cadeias de uma hélice de um DNA. De acordo com Chandar (2011), os mRNAs totalmente processados constituem apenas uma pequena proporção do RNA encontrado no núcleo, havendo, também, RNAs danificados e mal processados que acabam por ser degradados. Conforme Watson *et al* (2004), processo de síntese de proteínas a partir de mRNAs é conhecido como *tradução*.

O RNA ribossômico, ou rRNA, é o mais abundante dos quatro tipos e tem como função a síntese de proteínas. O rRNA é uma molécula intermediária, através da qual o DNA codifica para proteínas. Quatro tipos de rRNA são os maiores constituintes dos ribossomos, responsáveis pela síntese de proteínas (HARTL, 2011).

Os microRNAs são pequenos RNAs, com aproximadamente 22 nucleotídeos que regulam a expressão gênica ao nível pós-transcricional, ao se ligar nas regiões não traduzíveis, iniciadas no terminal 3' dos seus respectivos RNAs mensageiros. Os microRNAs podem se ligar completamente aos mRNAs, degradando as moléculas de RNA mensageiro, ou parcialmente, reprimindo a tradução do RNA mensageiro (SAYED et. al., 2013).

## 1.3 PROTEÍNAS

As proteínas são macromoléculas, também chamadas de polipeptídios, constituídas por uma ou mais cadeias de resíduos aminoácidos. Resíduos de aminoácidos são assim denominados pelo fato de, ao se ligarem uns aos outros, perderem uma molécula de água. Conforme Ramsden (2009), o nome deriva do ser mitológico *Proteus*, que podia assumir várias formas. As proteínas têm diversas funções dentro dos organismos vivos. Elas perfazem funções estruturais, catalíticas, realizam reações metabólicas, transporte de moléculas entre outras funções. As proteínas são componentes essenciais dos organismos e têm participação ativa em praticamente todos os processos celulares (JUNQUEIRA, 2011).

As proteínas são encontradas em variadas formas, como, por exemplo, pequenos polipeptídeos (no caso dos hormônios), em formas globulares, como a hemoglobina, em formas fibrosas, como a laminina, além de outras formas como as glicoproteínas e as proteínas de membrana (RAMSDEN, 2009).

A maioria das proteínas é composta por uma combinação única a partir de um conjunto de 20 L- $\alpha$ -aminoácidos. Cada conjunto de três nucleotídeos, chamados códons, em um DNA codifica para um aminoácido diferente. Como há 64 combinações possíveis de códons, diz-se que o código genético é redundante (ALBERTS 2008). As proteínas se organizam em quatro estruturas:

### 1.3.1 Estrutura primária

A estrutura primária é formada pela própria sequência de aminoácidos ao longo da cadeia polipeptídica. Com 20 aminoácidos possíveis, o número de polipeptídios que podem ser formados de comprimento  $n$  é  $20^n$  (KARP, 2008).

### 1.3.2 Estrutura secundária

A estrutura secundária é formada pelo arranjo espacial dos aminoácidos que estão próximos entre si na estrutura primária da proteína. A estrutura secundária descreve a conformação de porções da cadeia polipeptídica. Os polipeptídios podem formar estruturas conformacionais, conhecidas como  $\alpha$ -hélices e folhas- $\beta$ . As  $\alpha$ -hélices são trechos da proteína

em forma helicoidal e as folhas- $\beta$  são trechos da proteína na forma de *zigue-zagues* complementares, formando pranchas ou folhas (KARP, 2008).

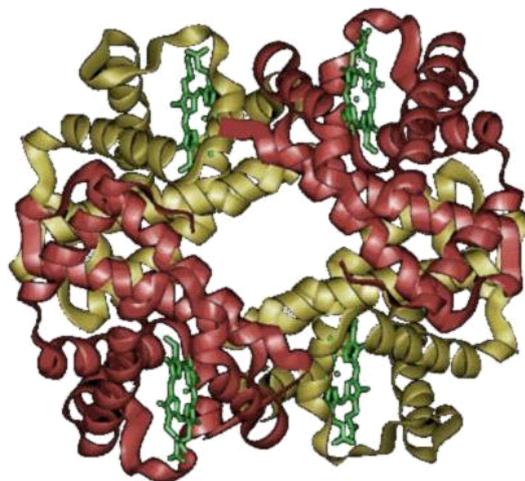
### 1.3.3 Estrutura terciária

A estrutura terciária descreve a estrutura da proteína inteira. Ela mostra a estrutura que envolve os diversos segmentos enovelados e suas interações, resultando em uma combinação de estruturas, conhecidas como  $\alpha$ -hélices, folhas- $\beta$  e outras cadeias estabilizadas por ligações de hidrogênio e ligações dissulfureto (KARP, 2008).

### 1.3.4 Estrutura quaternária

Quando mais de uma cadeia polipeptídica se junta para formar uma proteína, por exemplo a hemoglobina, tem-se uma estrutura quaternária. A Figura 2 ilustra a estrutura quaternária da hemoglobina, formada por duas cadeias polipeptídicas  $\alpha$  e duas cadeias polipeptídicas  $\beta$ .

Figura 2 - Estrutura quaternária da Hemoglobina



Fonte: WikimediaCommons

Estruturas de diversas proteínas podem ser obtidas em bancos de proteínas como, o *Protein Data Bank* (PDB). O PDB contém milhares de entradas que podem ser de estruturas de proteínas, estruturas de ácidos nucleicos, estruturas de carboidratos e modelos hipotéticos de proteínas (LESK, 2005).

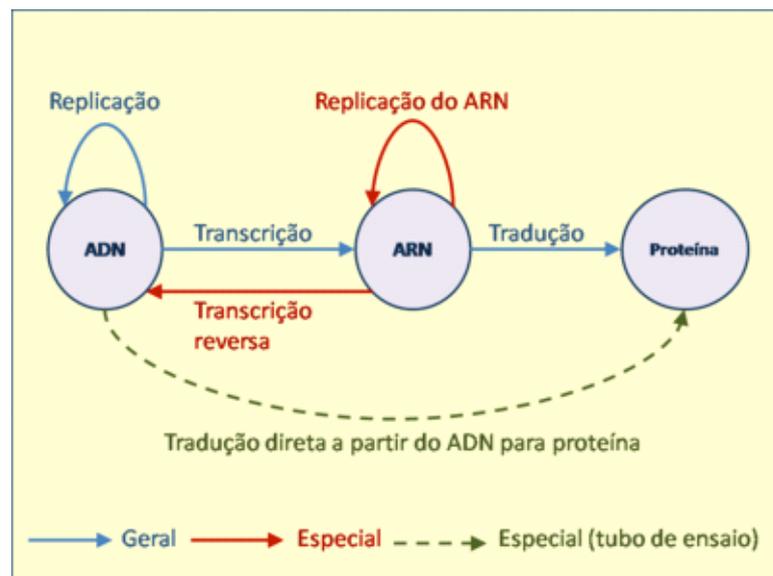
#### 1.4 O DOGMA CENTRAL DA BIOLOGIA MOLECULAR

O Dogma Central da Biologia Molecular é uma explicação do fluxo de informação genética dentro de um sistema biológico. Foi afirmado pela primeira vez em 1958 e publicado na revista *Nature* em 1970, por Francis Crick, um dos descobridores da estrutura em hélice do DNA. Conforme Crick (1970, p. 561),

O dogma central da biologia molecular trata da transferência detalhada resíduo-por-resíduo de informação sequencial. Ele afirma que tais informações não podem ser transferidas de volta a partir de proteínas, quer para proteínas ou para ácidos nucleicos.<sup>2</sup>

Com a evolução do conhecimento biológico posterior, algumas formas de transferência adicionais foram descobertas, que é o caso da síntese de DNA a partir de moldes de RNA que ocorrem em retrovírus, como o HIV. A Figura 3 ilustra as transferências atualmente conhecidas pela Biologia em seu estado da arte.

**Figura 3 - Esquema do Dogma Central da Biologia Molecular**



Fonte: Autor

De acordo com Hartl (2011), o termo dogma significa “conjunto de crenças”,. Essa denominação surgiu após a ideia ter sido estabelecida como uma teoria.

<sup>2</sup> Tradução do autor.

## 1.5 TÉCNICAS DE ANÁLISE DE DNA

A maioria das técnicas experimentais exige muitas cópias do mesmo fragmento de DNA. Um dos métodos mais utilizados é o PCR, reação em cadeia da polimerase (*polymerase chain reaction*). Neste método, fragmentos pequenos de DNA, com cerca de 100 a 500 nucleotídeos, são amplificados, produzindo um grande número de cadeias idênticas (HARTL, 2008). O PCR utiliza um molde de DNA com a região desejada que irá ser amplificada, um ou mais *primers* ou iniciadores, que identificam o trecho de DNA que será amplificado, uma DNA polimerase que irá sintetizar uma cópia do fragmento de ADN e quatro trifosfatos desoxirribonucleotídeos, dATP, dGTP, dCTP, e dTTP, que trabalham como blocos de construção da nova molécula de DNA (GRIFFITHS et. al., 2009).

Para ser possível estudar um gene específico de interesse, frequentemente é necessário cortar fragmentos do genoma do organismo e introduzir em um organismo hospedeiro, como, por exemplo, uma bactéria. Para esta tarefa, enzimas que funcionam como “tesouras”, chamadas enzimas de restrição, são usadas. Estas enzimas cortam ou clivam a fita dupla de DNA, reconhecendo sequências curtas de nucleotídeos específicas, denominadas sítios de restrição da enzima (HARTL, 2008).

Outra forma de utilização das enzimas de restrição é o emprego para sondar o estado de metilação do DNA (ALPHEY, 1997). Existe banco de dados, como o REBASE, onde se podem obter informações sobre uma enzima de restrição especificada (MARKELL e DARRYL, 2003). A Figura 4 ilustra um termociclador, aparelho utilizado na condução de PCR.

**Figura 4 - Termociclador, Aparelho utilizado para realizar uma PCR.**

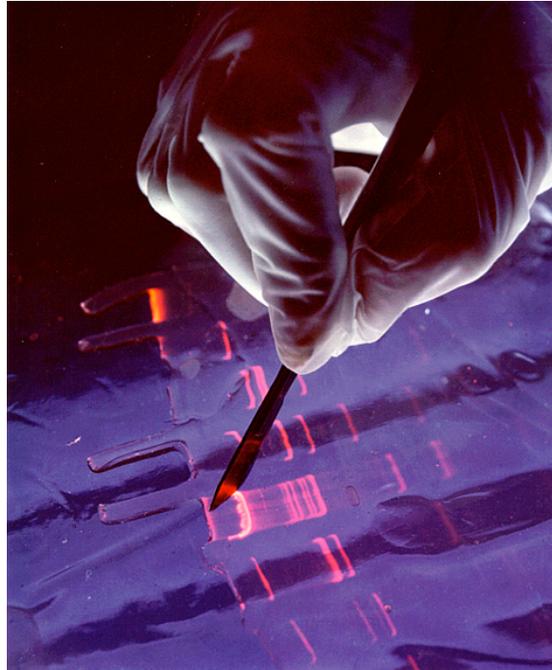


Fonte: Wikimedia Commons

Outra técnica utilizada com frequência para medir o comprimento do DNA é a *eletroforese em gel* (CONOLEY, 2008). Nesta técnica, o biólogo consegue separar os fragmentos em ordem de tamanho, porém não determinando a exata sequência de

nucleotídeos (JONES e PEVZNER, 2004). A Figura 5 ilustra a separação dos fragmentos em ordem de tamanho, a partir de deslocamentos a velocidades diferentes na camada de gel.

**Figura 5 - Separação por densidade de moléculas através da eletroforese em gel.**



Fonte: Creationwiki

Uma técnica de grande interesse é o sequenciamento de DNA ou RNA. Os métodos básicos de sequenciamento são os métodos desenvolvidos por Maxam e Gilbert (1977) e Sanger (1975). O primeiro exige rótulos radioativos e sua complexidade desencoraja o uso. O segundo, o método de Sanger, trabalha adicionando nucleotídeos modificados, chamados didesoxiribonucleotídeos, que barram o crescimento de um fragmento de DNA em replicação pela DNA polimerase após sua adição. Este método é amplamente utilizado. A técnica de Sanger é muito utilizada para o sequenciamento. Foi a técnica utilizada no projeto *Genoma Humano*. No sequenciamento a partir da técnica de Sanger, o DNA genômico é fragmentado em pedaços aleatórios, que são posteriormente clonados com a utilização de bactérias, constituindo uma biblioteca bacteriana. Em seguida, o DNA de cada clone bacteriano individual é sequenciado e a sequência é montada sobrepondo-se às regiões comuns do DNA com técnicas de alinhamento de sequências.

Entre os problemas encontrados na área de biologia molecular que são tratados pela biologia computacional encontram-se: o sequenciamento de genomas, localização de polimorfismos, identificação de genes e comparação de sequências. As técnicas de

alinhamento de sequências são usadas na resolução destes problemas. No capítulo a seguir serão descritos os principais algoritmos e técnicas de alinhamento.

## 2 ALINHAMENTO DE SEQUÊNCIAS

O objetivo deste capítulo é apresentar um dos problemas frequentemente tratados em bioinformática: o alinhamento de sequências. Pretende-se, neste capítulo, ilustrar as técnicas mais comuns e bem estabelecidas neste campo e que constam na grande maioria das ementas de cursos de bioinformática das universidades brasileiras.

### 2.1 BIOINFORMÁTICA

De acordo com Ramsden (2009), o termo bioinformática, inicialmente, referiu-se à aplicação da ciência da informação e à tecnologia nas ciências biológicas. Em uma definição posterior, o termo passou a denotar o uso de computadores para recuperar, processar, analisar e simular a informação biológica. De acordo com Geiger (2014), a bioinformática também é chamada de biologia computacional, embora alguns definam o termo bioinformática como restrito ao campo da biologia molecular.

O desenvolvimento e expansão das técnicas de sequenciamento do DNA tem permitido uma enorme disponibilidade de genomas em bancos públicos, resultando na necessidade de aumento de capacidade de armazenamento e técnicas de processamento adequadas para análise destes dados (JUNQUEIRA et. al., 2014). Em consequência disso, essas técnicas tornam-se áreas de estudo para cursos de biologia e afins. Atualmente, disciplinas de algoritmos e algoritmos de bioinformática têm sido ministradas em cursos de biologia e a tendência é de que sejam obrigatórios para todos os estudantes de biologia nas universidades de ponta (PEVZNER, 2008). Devido à extensão de suas aplicações, o alinhamento de sequências torna-se um processo de fundamental importância para a bioinformática, que é uma área de extenso uso e ensino (JUNQUEIRA et. al., 2014).

### 2.2 COMPARAÇÃO DE PARES DE SEQUÊNCIAS

Para a comparação de sequências de DNA, RNA e proteínas é uma das bases da bioinformática. (GIBAS, 2001, p.179). Segundo Sharma (2009), a comparação de sequências é, em si, um campo na ciência da computação. Ao se comparar sequências, podem-se analisar sequências de duas ou mais espécies de indivíduos diferentes. Quanto maior o número de sequências que são comparadas, maior é o tempo necessário para a obtenção do alinhamento (JUNQUEIRA et. al., 2014).

A comparação de múltiplas sequências está fora do escopo deste trabalho. A noção de similaridade de sequências é talvez o conceito mais fundamental na análise de sequências biológicas e não é uma tarefa trivial (BORODOVSKY e EKISHEVA, 2006). Conforme Setubal e Meidanis (1997) existem vários algoritmos de comparação, sendo os principais os descritos a seguir.

### 2.2.1 Comparação global

O algoritmo de alinhamento global tem como objetivo calcular a similaridade em toda a extensão de duas sequências, que podem ser DNA, RNA ou proteínas, dadas por meio de programação dinâmica. Saul B. Needleman e Christian D. Wunsch (NEEDLEMAN, S. AND WUNSCH, C., 1970) propuseram esse algoritmo em 1970, que serviu como base para outros algoritmos de sequenciamento posteriores, sendo também conhecido como algoritmo de *Needleman-Wunsch* (KORF ET AL., 2003).

Nesse algoritmo, se constrói uma matriz contendo, na primeira linha, uma das sequências e, na primeira coluna, a outra sequência a ser comparada. Lacunas ou *gaps* são inseridas de forma que as sequências resultantes tenham no final o mesmo comprimento (YE e TANG, 2008). Um *score* é calculado da seguinte forma: Para cada combinação igual se atribui um valor de casamento de padrões ou acerto (usualmente +1). Para cada combinação diferente se atribui um valor de não casamento de padrões (usualmente -1). Para o caso de se inserir uma lacuna tanto na primeira quanto na segunda sequência, se atribui um valor de penalização de lacuna (usualmente -2). O preenchimento da matriz se dá por programação dinâmica, sempre que se dispõe dos três valores (acerto, erro e lacuna) (SETUBAL, 1997).

Uma vez tendo definidos os valores de penalização e acerto, se faz o preenchimento da matriz, compondo-se os sucessivos valores de lacunas para a segunda linha e a segunda coluna. O Quadro 1 ilustra o estágio inicial de uma matriz preparada para a comparação entre as sequências AGGTG e GCTTG, considerando-se um valor de lacuna de -2.

**Quadro 1 - Matriz para comparação entre as sequências AGGTG e GCTTG.**

		<b>A</b>	<b>G</b>	<b>G</b>	<b>T</b>	<b>G</b>
	0	-2	-4	-6	-8	-10
<b>G</b>	-2					
<b>C</b>	-4					
<b>T</b>	-6					
<b>T</b>	-8					
<b>G</b>	-10					

Fonte: Autor

Iniciada a matriz, o algoritmo preenche as demais casas da matriz a partir da técnica de programação dinâmica, conforme o algoritmo descrito a seguir. Para cada casa da matriz  $a(i, j)$  ainda vazia, onde se conhece as posições  $a(i-1, j)$ ,  $a(i, j-1)$  e  $a(i-1, j-1)$ , se calcula o valor máximo de escore a partir da função:

$$a(i, j) = \text{máximo} \begin{cases} a(i, j-1) - 2 \\ a(i-1, j-1) + p(i, j) \\ a(i-1, j) - 2 \end{cases}$$

Onde  $p(i, j)$  é igual ao valor de acerto, se a linha casa com a coluna e, de outro modo, igual ao valor de não casamento de padrão, se os conteúdos da linha e da coluna forem diferentes.

Após este cálculo, o valor de  $a(i, j)$  é alimentado e o algoritmo prossegue preenchendo toda a matriz. O Quadro 2 ilustra a primeira posição calculada com o valor de não casamento de padrões, uma vez que na linha se tem o conteúdo A e na coluna o conteúdo G. Os três valores calculados são respectivamente -4, -1 e -4, de forma que o valor adotado é o maior dos três, -1.

**Quadro 2 - Matriz para comparação entre as sequências AGGTG e GCTTG após o primeiro preenchimento**

		<b>A</b>	<b>G</b>	<b>G</b>	<b>T</b>	<b>G</b>
	0	-2	-4	-6	-8	-10
<b>G</b>	-2	-1				
<b>C</b>	-4					
<b>T</b>	-6					
<b>T</b>	-8					
<b>G</b>	-10					

Fonte: Autor

Após esse passo, duas posições se tornam habilitadas para o cálculo, a saber, e Prosseguindo o algoritmo linha a linha, a posição seria calculada com o valor de acerto. Os

três valores calculados seriam, respectivamente, -3, -1 e -6. Portanto, o valor de seria igualmente -1. O Quadro 3 ilustra este preenchimento.

**Quadro 3 - Matriz para comparação entre as sequências AGGTG e GCTTG após o segundo preenchimento**

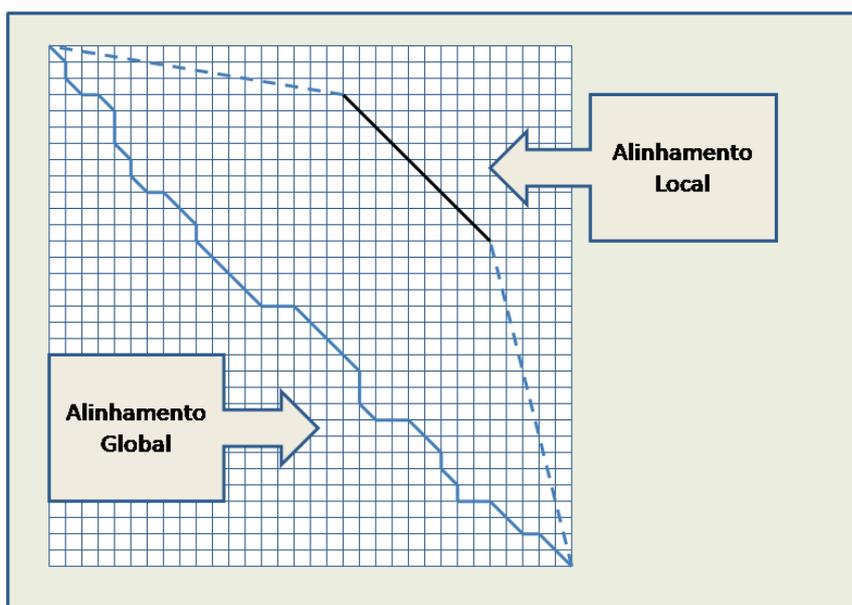
		A	G	G	T	G
	0	-2	-4	-6	-8	-10
G	-2	-1	-1			
C	-4					
T	-6					
T	-8					
G	-10					

Fonte: Autor

### 2.2.2 Comparação local

Frequentemente similaridades biológicas significativas estão presentes em partes de fragmentos de DNA, mas não estão presentes em outras. Para tal, o algoritmo de comparação local foi desenvolvido (PEVZNER, 2000). Nessa técnica, os *escores* de alinhamento entre duas subsequências,  $v$  e  $w$ , por exemplo, podem ser maiores que o *escore* obtido da totalidade das sequências  $v$  e  $w$  (JONES e PEVZNER, 2004). A Figura 6 ilustra uma comparação entre o alinhamento local e o alinhamento global.

**Figura 6 - Comparação entre alinhamento local e alinhamento global.**



Fonte: Adaptado de Jones e Pevzner, 2004

O algoritmo de alinhamento local descrito neste trabalho foi, primeiramente, proposto por Smith e Waterman e é uma modificação bastante simples do algoritmo de Needleman-Wunsch. As mudanças consistem das seguintes alterações:

- As arestas da matriz são inicializadas em zero, ao invés de penalidades para lacunas cumulativas;
- O escore máximo nunca é menor que zero;
- O alinhamento traça uma rota reversa, a partir do mais alto *escore* na matriz, em vez de ser a partir da borda da matriz e termina no escore zero, no lugar de terminar no início da matriz (KORF ET AL., 2003).

Essas mudanças simples têm o efeito de alterar, substancialmente, o comportamento do algoritmo. Segundo Setubal (1997, p. 55), a função de *escore* do algoritmo de alinhamento local seria modificada e teria a forma:

$$a(i, j) = \text{máximo} \left\{ \begin{array}{l} a(i, j - 1) - 2 + \text{penalização lacuna} \\ a(i - 1, j - 1) + p(i, j) \\ a(i - 1, j) + \text{penalização lacuna} \\ 0 \end{array} \right.$$

O maior valor de  $a(i, j)$ , sobre o grafo total, representa o melhor alinhamento local para as sequências sendo comparadas (JONES e PEVZNER, 2004). Entre as vantagens do alinhamento local sobre o alinhamento global, estão a aplicação de comparação de sequências entre dois organismos, distantemente relacionados, que compartilham apenas poucos domínios não contíguos e a análise de elementos repetidos dentro de uma única sentença (CLAVERIE e NOTREDAME, 2003). O alinhamento local tem proporcionado a descoberta de relacionamentos entre espécies que, de forma global, tem pouca similaridade (WATERMAN, 2000).

### 2.2.3 Comparação semi-global

Segundo Setubal (1997), em uma comparação semi-global, os alinhamentos são pontuados, ignorando-se alguns dos espaços finais presentes nas sequências. O objetivo desse

algoritmo é evitar a inserção de um grande número de espaços internos à cadeia. Por exemplo, no alinhamento da Figura 7, existe uma considerável extensão de lacunas.

**Figura 7 - Alinhamento local, com lacunas**

<p>CAGC A – C T T G GAT T C T C GG  CAGC -----G – T----- GG</p>
---

Fonte: Setubal, 1997

Não penalizando os espaços antes e depois da cadeia analisada, resulta em um alinhamento compacto como exibido na Figura 8.

**Figura 8 - Alinhamento local, com lacunas**

<p>CAG CA – CTTGGATTCTCGG  ---CAGCGTGG-----</p>
---

Fonte: Setubal, 1997

## 2.2.4 Outras comparações

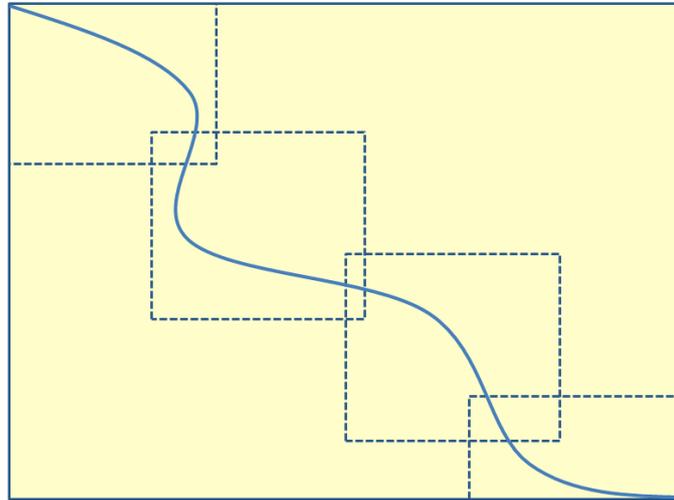
Outros algoritmos para comparação foram propostos para casos específicos, mas que estão além do escopo da ferramenta proposta, uma vez que não são abordados em disciplinas de cursos de graduação de biologia. Entre esses, estão os algoritmos de alinhamento rápido global de sequências, alinhamento emendado (*spliced alignment*), alinhamento de sequências eficiente em termos de espaço (*space-efficient sequence alignment*), alinhamento segmentar e alinhamento em ordem parcial.

### 2.2.4.1 Alinhamento rápido global de sequências

O algoritmo de alinhamento rápido global de sequências, que deriva do inglês *Fast Global Sequence Alignment*, procura aumentar a velocidade dos algoritmos de programação dinâmica com estratégias heurísticas, tais como métodos de encadeamento (YE e TANG, 2008). O procedimento de encadeamento foi proposto pela primeira vez por Wilbur e Lipman (1983) e foi implementado nos conhecidos programas FASTA. Essa estratégia procura alinhar grandes sequências em subconjuntos, encadeados e ligados por âncoras, de forma a se concentrar na diagonal principal, onde o alinhamento tende a se dar. A Figura 9 ilustra o

alinhamento com quatro subconjuntos encadeados. Nas áreas sobrepostas encontram-se as âncoras que ligam um subconjunto ao outro.

**Figura 9 - Encadeamento no Alinhamento Rápido Global de Sequências**

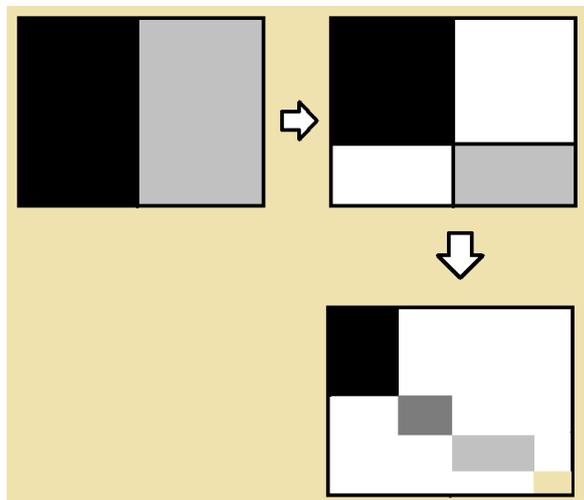


Fonte: Autor

#### 2.2.4.2 Alinhamento de seqüências eficiente em termos de espaço

O algoritmo de alinhamento de seqüências eficiente em termos de espaço, do inglês *Space-efficient Sequence Alignment*, é semelhante ao alinhamento rápido global de seqüências, mas procura gerar os segmentos em uma estratégia recursiva. Ele utiliza uma estratégia de dividir para conquistar, calculando pontos centrais na matriz e subdividindo o problema em dois, recursivamente decrescendo por um fator de 2. (PEVZNER, 2000). A Figura 10 ilustra as fases recursivas reduzindo a área de busca utilizada para o alinhamento.

**Figura 10 - Fases recursivas no alinhamento de seqüências eficiente em termos de espaço**



Fonte: Autor

#### 2.2.4.3 Alinhamento emendado

O algoritmo de alinhamento emendado, do inglês *Spliced Alignment*, procura considerar o *splicing* alternativo e, desta forma, esle diferencia os éxons dos íntrons. Os éxons são os trechos de DNA que são efetivamente traduzidos em proteínas, ao passo que os íntrons são descartados durante o processo de tradução. Este algoritmo se aplica somente às proteínas, uma vez que está considerando o produto da tradução (YE e TANG, 2008).

#### 2.2.4.4 Alinhamento segmentar

O algoritmo de alinhamento segmentar, do inglês *Segmental Alignment*, procura considerar uma ideia semelhante a do alinhamento emendado. Da mesma forma, esse algoritmo se aplica somente às proteínas. Nesse algoritmo, a comparação não se dá pelos aminoácidos individuais, mas para segmentos preditos, contendo estruturas mínimas que são encontradas em proteínas não homólogas (YE e TANG, 2008).

#### 2.2.4.5 Alinhamento em ordem parcial

O algoritmo em ordem parcial, do inglês *Partial Order Alignment*, procura trabalhar com alinhamento de sequências múltiplas, buscando formar uma ordem parcial entre elas. (YE e TANG, 2008).

No próximo capítulo serão tratadas as metodologias utilizadas nesse trabalho. Será descrita a modelagem do fluxo de dados, o modelo entidade de relacionamento e os diagramas UML.



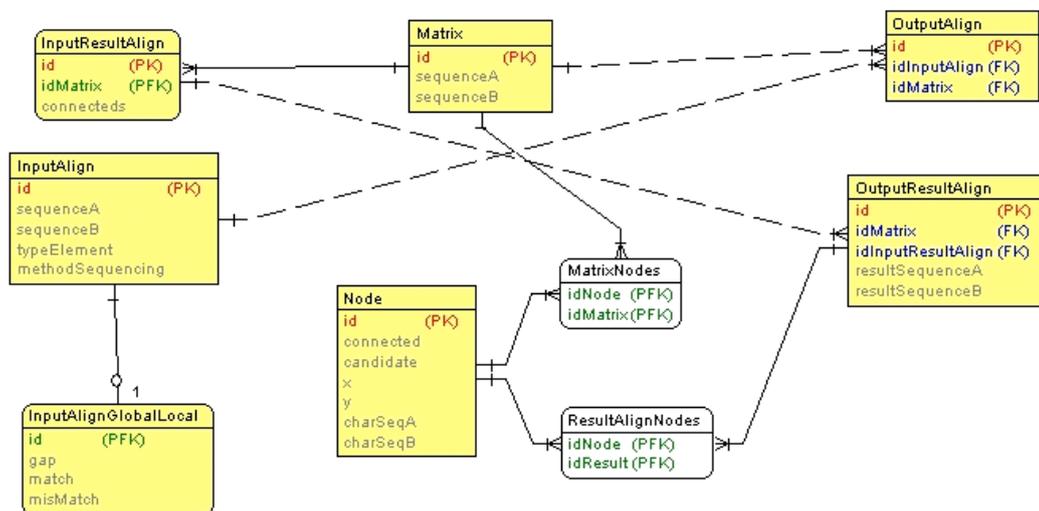
### 3.2 DIAGRAMA DE ENTIDADE-RELAÇONAMENTO

O diagrama de entidade-relacionamento descreve a estrutura geral da representação da estrutura do banco de dados de forma gráfica (CHEN, 1990). A Figura 12 ilustra o modelo Entidade Relacionamento elaborada para o *software* proposto. As entidades são representadas em caixas na cor amarela, contemplando os atributos. Os relacionamentos são representados pelas linhas entre as entidades.

Figura 12 - Modelo ER

[1.1]

[2.1]



Fonte: Autor

### 3.3 UML- UNIFIED MODELING LANGUAGE

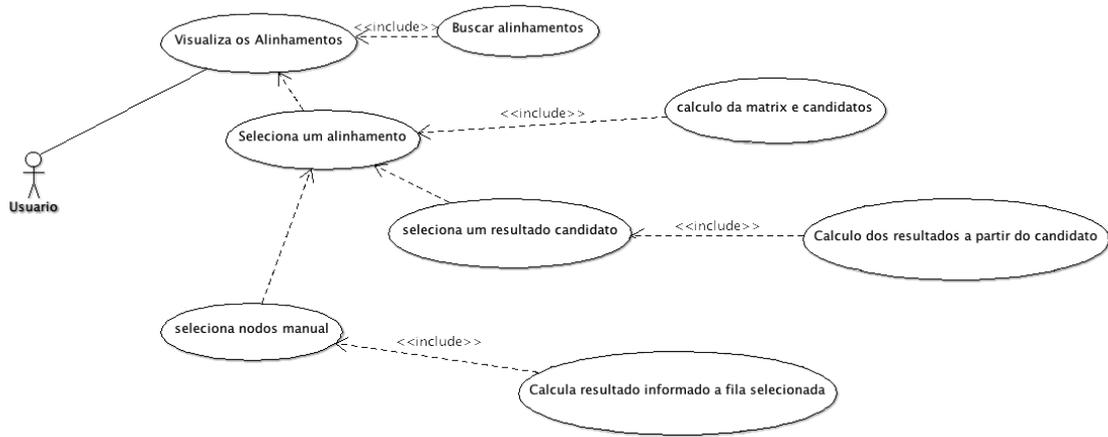
A UML ou *Unified Modeling Language* é um conjunto de notações gráficas padronizados, apoiadas por um metamodelo único, objetivando apoiar a descrição e o projeto de sistemas de *software*, em especial, os que utilizam o desenvolvimento orientado a objetos (FOWLER, 2005).

#### 3.3.1 Caso de Uso

O estudo com o diagrama de caso de uso tem como princípio o entendimento das iterações entre o sistema e o usuário. Na Figura 13 está representada a interação entre a ferramenta de alinhamento e o usuário, demonstrando as 3 principais iterações da ferramenta.

As interações não são simples entre as operações e a ferramenta, pois a sua complexidade se encontra na elaboração de cálculos e definições programadas internamente.

Figura 13 - Caso de Uso

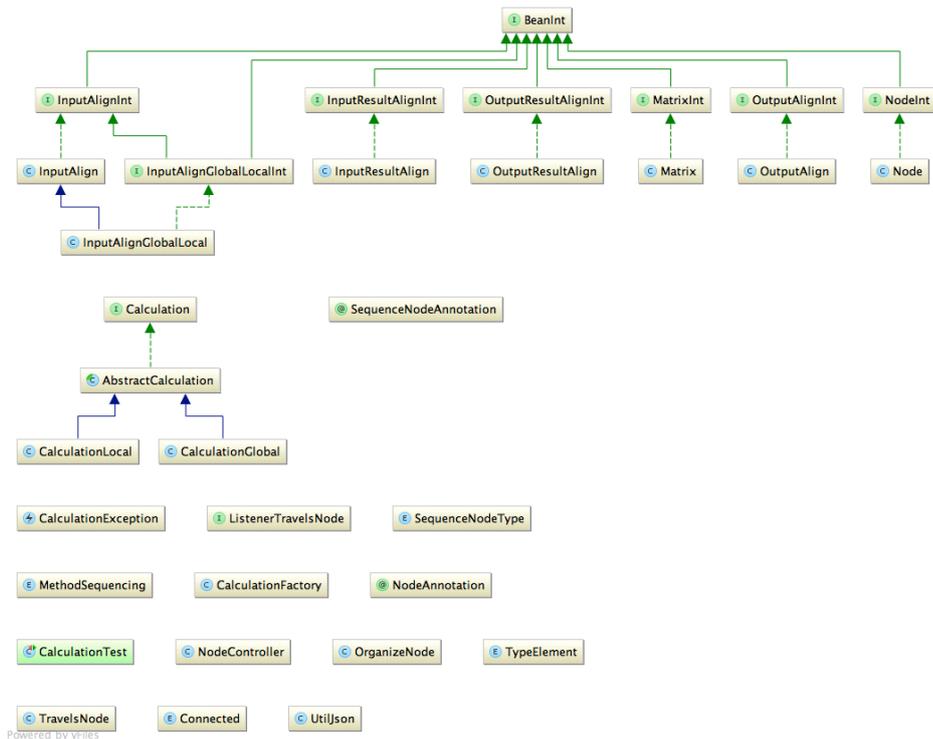


Fonte: Autor

### 3.3.2 Diagrama de Classe

O estudo com o diagrama de classe é a representação da estrutura na forma gráfica das classes implementadas e/ou a serem desenvolvidas. Na Figura 14 há o modelo gráfico das classes implementadas para realizar o processo de alinhamento.

Figura 14 - Diagrama de Classe



Fonte: Autor

### 3.4 POLÍTICA OPEN SOURCE

Nesta seção será descrita a disponibilização do software na plataforma *open source*. Além disso, o uso de método e ferramentas *open source* para controle de versão, testes unitários e distribuição da ferramenta.

#### 3.4.1 Fontes Colaborativos

O uso de ferramentas colaborativas para distribuição dos fontes vem crescendo cada dia mais. Entre os mais utilizados estão o *SourceForge* (<http://sourceforge.net>) e o *Github* (<http://github.com>).

##### 3.4.1.1 SourceForge

O SourceForge está entre os primeiros a oferecer o serviço de gerenciamento de projetos abertos. O controle de versão feita pelo SourceForge pode ser em subversion ou git, atualmente o projeto de DNA-Sequence está disponível em git.

##### 3.4.1.2 Github

O Github é uma ferramenta colaborativa de código fontes no formato livre ou privado. Muitas das ferramentas livres utilizadas comercialmente estão disponíveis na forma colaborativa no site com o Rails, jQuery e Node.js. O controle de versão realizado pelo o Github é o git.

Segundo Loeliger e McCullough (2012), ao pensar em um cenário minimalista, o Github é categorizado como apenas um local de hospedagem de projetos *open source*, no entanto, o conceito de repositório não é algo novo e tem sido aberto caminho por outras hospedagens, como *SourceForge* e *Google Code*, entre outros, apoiados pela força de seus usuários. Mas o Github tem uma abordagem diferente que estende o conceito de hospedagem de projetos *open source* com contribuição da comunicação. O Github oferece aspectos sociais na *web*, vistos no *Twitter*, *Facebook* e outras redes sociais.

### 3.5 CONTROLE DE VERSÃO

O uso de ferramentas de controle de versão é algo muito importante, tanto na conservação das versões, conforme novas implementações surgem, como quando há a necessidade de realizar e compartilhar os fontes e contribuições.

A ferramenta de controle de versão que foi escolhida foi o *Git* (<http://git-scm.com/>), por se tratar de uma ferramenta pratica e de fácil adaptação.

### 3.6 TESTE UNITÁRIO

O desenvolvimento de uma ferramenta requer algum parâmetro de validações que resultado não apresente distorções. Como uma boa prática e como forma de amenizar os erros apresentados na ferramenta, é adotado o uso do teste unitário.

O teste unitário é um teste de entrada e saída de dados, de acordo com a entrada é validada a própria saída e, caso ocorra o contrário, gera erro no processo.

O *JUnit* é uma ferramenta *open source*, utilizada para realizar teste unitário na plataforma Java. A elaboração de caso de teste feita pelo *JUnit* é relativamente fácil e importante. É uma boa pratica de desenvolvimento de *software*, pois possui integração com o *Maven* para realizar a automatização dos testes definidos.

O *JUnit* é uma ferramenta de teste unitário na plataforma *javascript* e tem um papel importante para realização de testes de interface e de processo *javascript*. Possui integração com o *Gruntjs* para automatizar os testes definidos.

### 3.7 FERRAMENTA DE AUTOMAÇÃO E GERENCIAMENTO DE DEPENDÊNCIAS

O processo de desenvolvimento, muitas vezes, acaba se alongando quando há uma necessidade de processo manual. Assim, a melhor forma de aperfeiçoar o trabalho manual é a criação de meios de automatizar tarefas repetitivas.

O *Maven* é uma ferramenta de automação de compilação, possui controle de dependência de pacotes, através de um arquivo *xml* na pasta principal do projeto com o nome de *pom.xml*. Esse *xml* apresenta parâmetros e definição das informações de pacotes e o

resultado a ser gerado pelo projeto, como, por exemplo, a execução de testes unitários, criação de pacotes *.jar* e, até mesmo, o *deploy* de um projeto para o servidor de produção.

O *Gruntjs* se trata de uma ferramenta de automatização de tarefas. Tem seu conceito semelhante ao *Maven*, com o foco em *javascript*, tanto em *Front-end* quanto *Back-end*. A sua utilização é muito relevante no agrupamento de arquivo *javascript*, tendo, assim, apenas um arquivo para distribuição do projeto.

O *Brower* é uma ferramenta de controle de dependência totalmente voltada à *Front-end* (interface com o usuário), que auxilia na organização dos pacotes necessários ao projeto, como, por exemplo, no projeto DNA-Sequence, em que foram utilizados outros projetos, como jQuery, i18next, mootools, entre outros. Para controlar esses projetos e suas versões tornam-se um pouco complicadas a organização e atualização da ferramenta, por isso, para melhorar este processo, adotou-se a ferramenta *Brower*.

### 3.8 NODE.JS

Basicamente, o *Node.js*, interpretador *javascript* no servidor, possibilita trabalhar com biblioteca *javascript* não somente com interface. O *Node.js* possui um grande poder de estabilidade de servidor *web*, sendo, assim, um ponto importante no uso da ferramenta.

Com o crescimento do *Node.js*, surge, cada vez mais, novas ferramentas que se utilizam da tecnologia, através de um gerenciador de módulos em *Node.js*, chamado de NPM (*Node Packaged Modules*), incluindo diversas ferramentas importantes, como: *gruntjs*, *phonegap*, *Brower*.

O NPM tem semelhança ao repositório *Maven*, com a diferença de que os projetos *Maven*, em sua maioria, é implementado em *Java*, já o módulo NPM na é implementado em *javascript*.

### 3.9 WEB E NODEWEBKIT

Atualmente, as ferramentas que apresentam versões *desktop* sem uma disponibilidade do uso *web*, muitas vezes, acabam dificultando o seu uso, quando usamos uma ferramenta que necessita de instalação de outros *frameworks* para executar.

Dessa forma, surgiu a necessidade de desenvolver a ferramenta totalmente *web*, assim dependendo do *browser* de navegação.

O uso da tecnologia *NodeWebkit* se trata do desenvolvimento com *HTML*, *JavaScript* e *CSS*, gerando executáveis para as principais plataformas *desktop* (*Window*, *OSX* e *linux*).

Os mesmos algoritmos desenvolvidos na versão *web* são reaproveitados na versão *NodeWebkit*, havendo, assim, uma disponibilidade da ferramenta *on-line* quando *off-line*.

### 3.10 GWT - JAVA

O *GWT* (*Google Web Toolkit*) é um *framework* desenvolvido em *Java* para facilitar o desenvolvimento da camada de interface *web*, podendo utilizar a linguagem *Java* na construção da interface, convertendo, assim, elementos do processo de alinhamento desenvolvidos na linguagem *JavaScript* no ambiente *web*. Como Hanson e Tacy (2007, p. 4) descrevem.

Em maio de 2006, o Google lançou o Google Web Toolkit (GWT), um conjunto de ferramentas de desenvolvimento, programação de utilitários e widgets, permitindo que você crie aplicações ricas para internet de forma diferente do que você poderia ter feito antes. A diferença entre GWT e todas essas outras estruturas, é que com o GWT você escreve seu código do lado do navegador em Java em vez de JavaScript. Para aqueles de nós que dependem de Java como uma ferramenta confiável esta é realmente uma diferença monumental sobre JavaScript tradicional de codificação.<sup>3</sup>

### 3.11 HTML5/CSS3 - JQUERY

O *HTML*, durante sua história, passou por diversas melhorias e padronizações. Com o surgimento do modelo padrão *HTML5*, buscou-se uma maior interatividade entre a página *web* e o usuário. O uso de *jQuery* é um grande facilitador para o desenvolvimento da camada de interface com métodos, otimizada e padronizada, para não haver derivações de código para determinado browser, ambiente ou versão.

### 3.12 SVG -SCALABLE VECTOR GRAPHICS

O *SVG* (*Scalable Vector Graphics*) é um gráfico de vetores escaláveis. Sua estrutura é um arquivo *XML*, que define a representação na informação gráfica. O *HTML5* possui o

---

<sup>3</sup> Tradução do autor.

suporte ao uso de imagem *SVG* no *browser* e interação com a *interface web*, sendo possível, assim, elaborar imagens dinâmicas com o uso de comando *javascript*.

A construção da matriz presente no *software DNA-Sequence* foi montada em *SVG* com interações e métricas, através de comando *javascript* e modelos *CSS*. A matriz apresenta uma forma melhor de trabalhar quando utilizamos imagens em *SVG* através de *XML*.

### 3.13 DADOS ARMAZENADOS

Quando usamos ferramenta *online*, precisamos armazenar dados no ambiente do cliente. No *browser* antigo havia uma limitação de uso de *cookies* e pouca interação com o usuário. Com o uso do *LocalStorage*, podemos arquivar dados no *browser* do usuário, diminuindo a interação com o servidor e o cliente, tornando o processo menos dependente do servidor.

### 3.14 INTERNACIONALIZAÇÃO E LOCALIZAÇÃO

A *i18next* é uma ferramenta desenvolvida em *Javascript* que identifica a região definida ao *browser* e apresenta as informações da interface do idioma conFigurado ao usuário, mapeando os textos e informações apresentadas no *software DNA-Sequence*. Foi realizada a internacionalização de toda a interface nos idiomas português e inglês. Caso seja necessário incluir uma nova tradução do *software*, basta criar um novo arquivo de parâmetro na linguagem escolhida.

O exemplo de arquivo de parâmetros entre o idioma português e o inglês em anexo no **Apêndice C**.

### 3.15 BOOTSTRAP

O *Bootstrap* tem sua origem no *Twitter*. Com uma interface moderna e amigável, tornou-se cada vez mais usada na construção de *interface web* e o grande potencial presente na ferramenta está aliado à facilidade de construção de *interface* rica. O mesmo vem evoluindo e trazendo diversas interações e tecnologias presentes nos conceitos *HTML5* e *CSS3*.

### 3.16 HANDLEBARS

A ferramenta *Handlebars* é uma evolução da ferramenta *Mustache*, funcionando como um analisador semântico eficiente, com o uso de algumas notações e encapsulamento entre pares de chave, como, por exemplo: `<h1>{{titulo}}</h1>`.

A ferramenta foi utilizada para implementação de parâmetro de configuração de cores e modelos *CSS* no software *DNA-Sequence*, criando predefinições do modelo *CSS*, sendo possível ser alterado pelo usuário.

No próximo capítulo serão tratadas questões sobre a ferramenta de alinhamento, tratando da interação entre as *interfaces* e a codificação em *Java* e *JS*, exemplificando e demonstrando os modelos de alinhamento apresentados nos capítulos anteriores.

## 4 A FERRAMENTA DE ALINHAMENTO

Neste capítulo será descrita a tecnologia utilizada no desenvolvimento e publicação da ferramenta de alinhamento.

A ferramenta de alinhamento disponível como uma ferramenta *open source*, código fonte aberto para contribuição, que tem sua página publicada do *Github SourceForge*, possui uma interface simples e acessível. O acesso à ferramenta *web* está disponível em <http://dna-sequence.sf.net/app> ou <http://dna-sequence.github.io/app>, desde o mês de maio de 2014.

### 4.1 INTERFACE DNA-SEQUENCE

O *DNA-Sequence* tem sua interface inicial apresentada na Figura 15. Possui a listagem de alinhamentos cadastrada e três botões: novo (*new*), abrir (*open*) e o deletar (*delete*). A *interface* tem internacionalização de palavras reservadas, atualmente configurada para dois idiomas: português e inglês.

Figura 15 - Interface inicial



#	Align	Date	Type
<input type="checkbox"/> 0	1: ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC 2: CTGCAGGAACCTTCTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCAGGCAAG	07/06/2014	GLOBAL
<input type="checkbox"/> 1	1: ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC 2: CTGCAGGAACCTTCTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCAGGCAAG	23/06/2014	LOCAL
<input type="checkbox"/> 2	1: TTTTGAGGTCATCTGTTGTTCAAACCTCWTATGGTGAATTTTTG 2: TTGGAAATTATTGCTGAGGTTGTTGATTCCAACATATTGTTTT	26/06/2014	GLOBAL
<input type="checkbox"/> 3	1: ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC 2: CTGCAGGAACCTTCTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCAGGCAAG	26/06/2014	GLOBAL
<input type="checkbox"/> 4	1: ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC 2: CTGCAGGAACCTTCTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCAGGCAAG	26/06/2014	LOCAL
<input type="checkbox"/> 5	1: ACACACTA 2: AGCACACA	15/09/2014	LOCAL

Fonte: Autor

A listagem de alinhamentos contém algumas informações, como: as sequências informadas, a data de criação, método utilizado para o cálculo da matriz e um índice para ordenar os alinhamentos na lista.

#### 4.1.1 Novo alinhamento

Para criar um novo alinhamento é necessário clicar no botão “Novo” (*New*). Ao clicá-lo, irá apresentar uma interface é possível cadastrar os dados necessários para implementação dos algoritmos de alinhamento, que são: nucleotídeos ou proteínas, a sequência A, a sequência B, os métodos de alinhamento, os valores de *gap*, *match*, *mismatch* e a exceção de valores *mismatch*, detalhados nesse capítulo.

A Figura 16 apresenta o protótipo realizado em papel que exemplifica o funcionamento de inclusão de um novo alinhamento.

Figura 16 – Protótipo de cadastro de novo alinhamento

The image shows a hand-drawn prototype of a sequence alignment registration interface. It consists of the following elements:

- Two input fields for sequences: "SEQ A" and "SEQ B", both containing the text "AAAA".
- A "TYPE" section with two radio button options: "GLOBAL" and "LOCAL".
- Under the "GLOBAL" option, there is a box containing three input fields labeled "GAP:", "MATCH:", and "MISMATCH:".
- Under the "LOCAL" option, there is a box containing three input fields labeled "GAP:", "MATCH:", and "MISMATCH:".
- At the bottom of the interface, there is a large button labeled "RUN".

Fonte: Autor

A Figura 17 esta a interface de inclusão de um novo alinhamento presente na ferramenta DNA Sequence, todos campos previsto no prototipo.

**Figura 17 - Cadastro de novo alinhamento**

New Align

NUCLEOTIDE  PROTEIN

Sequence A  
ACAAGATGCCATTGTCCCCCGGCCTCTGCTGCTGCTCTCCGGGGCCACGGC

Sequence B  
CTGCAGGAACCTTCTTCTGGAAGACCTTCTCCTCTGCAAATAAACCTCACCCATG^

METHOD

Global  Local  Semi-global

Gap Penalty

Match

Mismatch

Mismatch A:  B:

Fonte: Autor

#### 4.1.1.1 Nucleotídeos ou Proteínas

Se a sequência é baseada em Nucleotídeos ou Proteínas é a primeira informação que deve ser cadastrada. Na Figura 16 é apresentada a versão no idioma inglês, tendo, então, como opção: *Nucleotide* ou *Protein*.

#### 4.1.1.2 Sequência A e B

A informação necessária para realizar o alinhamento dos mesmos, na Figura 16 está apresentada como *sequence A* e *B*.

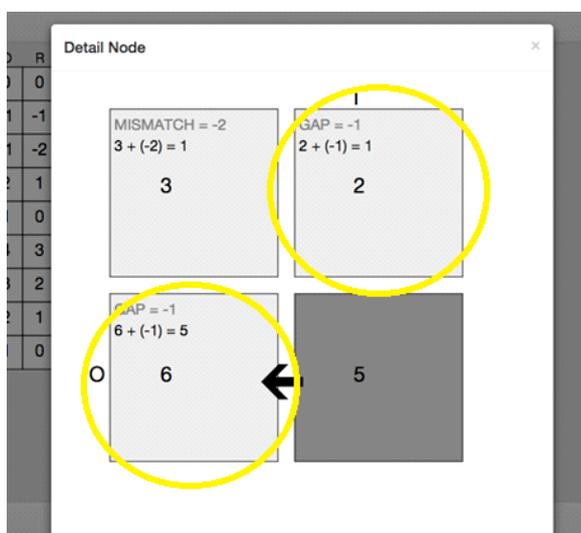
#### 4.1.1.3 Método (Global, Local, Semi-global)

A partir dos métodos descritos nos capítulos anteriores é possível analisar o processo de alinhamento, de acordo o método solicitado, podendo ser informado apenas um método ou mais.

#### 4.1.1.4 Gap Penalty

Quanto ao valor necessário para realizar o cálculo da matriz de alinhamento, ao analisar a Figura 18, é possível verificar que seus vizinhos da esquerda e superior, destacados pelo círculo amarelo, realizam a soma a partir do valor informado ao *gap*.

**Figura 18 - Detalhe de um nodo selecionado destacando Gap**

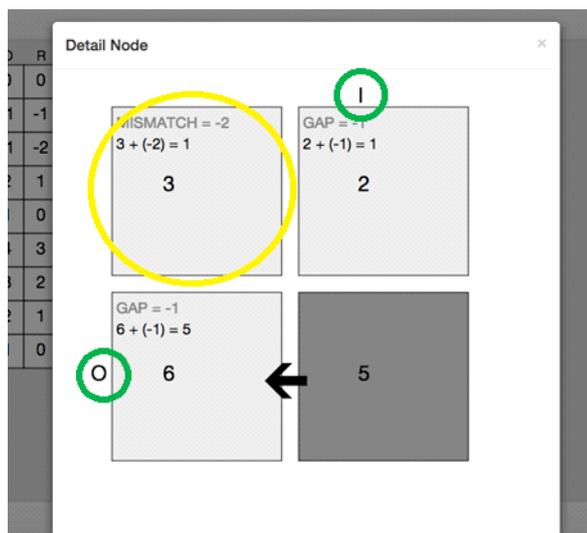


Fonte: Autor

#### 4.1.1.5 Match e Mismatch

No que se refere ao valor necessário para realizar o cálculo da matriz de alinhamento, ao analisar a Figura 19, é possível verificar que seu vizinho na diagonal, destacado pelo círculo amarelo, realiza a soma. Caso a ligação da matriz, destacado pelo círculo verde, seja igual, é utilizado o valor *match*, caso contrário é utilizado o valor *mismatch* ou a exceção.

**Figura 19 - Detalhe de um nodo selecionado destacando Mismatch**



Fonte: Autor

#### 4.1.1.6 Exceção Mismatch

Ao analisar a Figura 19, é possível verificar que a ligação resultou uma ligação *mismatch*. Neste caso é utilizado o valor do campo *mismatch*, no entanto, é possível realizar combinações diferentes de *mismatch* com valores diferenciados do valor definido no campo *mismatch*. Essa exceção pode se incluir na Figura 17, basta informar a ligação do valor da sequência de A e B, o valor de *mismatch* com exceção e clicar em *add*.

Ao salvar o alinhamento, ele estará presente, na lista de alinhamentos apresentados na Figura 15.

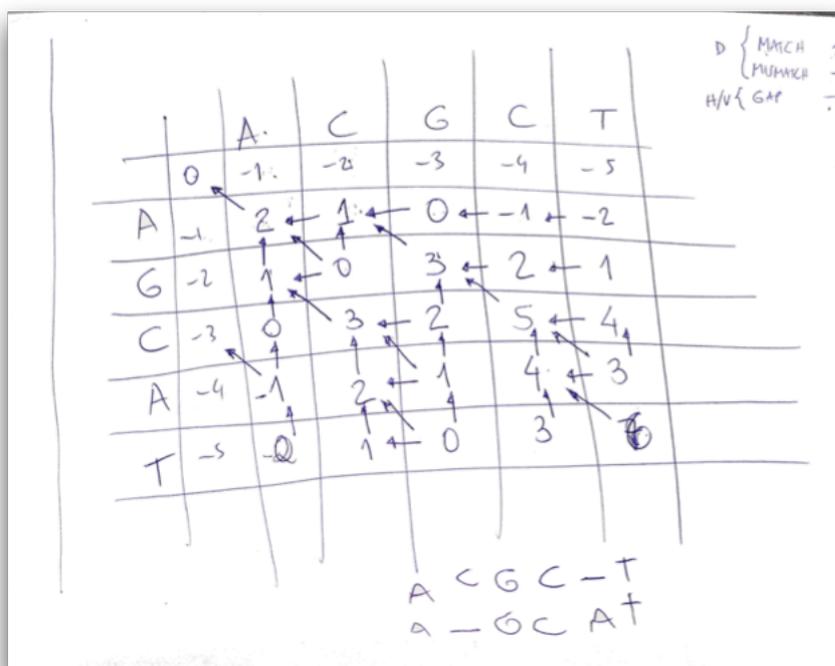
#### 4.1.2 Deletar alinhamento

Para selecionar um alinhamento, basta clicar em qualquer alinhamento ou em seu *checkbox*. Caso queria excluir é necessário selecionar o alinhamento e clicar no botão "Deletar" (*Delete*).

#### 4.2 ABRIR UM ALINHAMENTO

A Figura 20, apresenta um protótipo de forma que seja possível visualizar a estrutura da matriz e seu resultado facilmente. Com base no prototipo foi criar uma interface apresentada na Figura 21.

Figura 20 – Protótipo de exemplo de alinhamento global



Fonte: Autor

Quando selecionado um alinhamento e, ao clicar em “Abrir” (*Open*). Na Figura 15 é apresentada uma interface. Na Figura 19 há os botões:

- Voltar (*Back*), para retornar para listagem de alinhamentos;
- Parâmetros (*Parameter*) exibe uma interface para alterar as cores e padrão da representação da matriz do alinhamento;
- Escala (*Scale*) ao clicar é apresentado uma lista de 50%, 100%, 150% e 200%, assim aplicando uma ampliação ou redução visual da matriz.

Ao lado direito, encontramos uma lista de alinhamentos pré-definidos e um botão “Novo Manual” (*New Manual*). Quando clicado em um dos alinhamentos pré-definidos, é apresentado o alinhamento em cinza e seu resultado na área inferior da interface.

A matriz apresentada na parte central da interface é a representação das sequências informadas no cadastro, acompanhada da implementação dos algoritmos. Ao analisar a matriz, é possível verificar que na horizontal superior está presente a “Sequência A” (*Sequence A*), dividindo cada valor de um nodo da matriz. Ao início dos valores da sequência é incluso um valor em branco. Na vertical é realizado o mesmo procedimento para a “Sequência B” (*Sequence B*).

Figura 21 - Exemplo de alinhamento global

	B	I	O	I	N	F	O	R	M	A	T	I	C	A	
0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	
B	-1	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
I	-2	1	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
O	-3	0	3	6	5	4	3	2	1	0	-1	-2	-3	-4	-5
L	-4	-1	2	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6
O	-5	-2	1	4	3	2	1	4	3	2	1	0	-1	-2	-3
G	-6	-3	0	3	2	1	0	3	2	1	0	-1	-2	-3	-4
I	-7	-4	-1	2	5	4	3	2	1	0	-1	-2	1	0	-1
A	-8	-5	-2	1	4	3	2	1	0	-1	2	1	0	-1	2

Fonte: Autor

Os valores da matriz são o resultado do cálculo dos algoritmos, citado nos capítulos anteriores. A partir dos nodos representados na matriz, pode ser visualizado, de uma forma mais detalhada, o algoritmo, representado na Figura 19. Para obter o detalhamento do nodo em questão é necessário clicar com o botão direito sobre o mesmo. A execução do processo de alinhamento é realizada a entrada da geração da matriz, conforme pode ser conferido em anexo no **Apêndice A**.

Quando analisamos o protótipo na Figura 20, o mesmo apresenta mais informações que a interface implementada na figura 21, para melhorar o entendimento do processo, foi implementado uma detalhamento dos nodos de forma individual apresentando no protótipo na Figura 22.

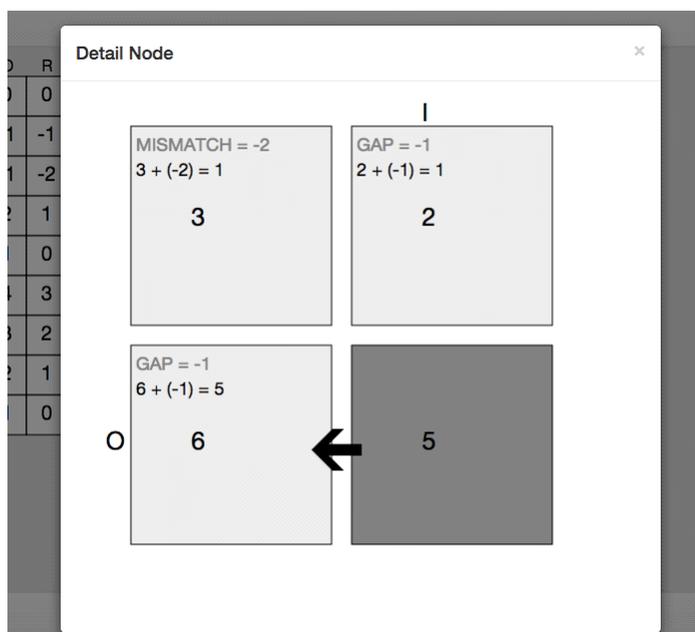
Figura 22 – Protótipo detalhe de um nodo selecionado

		A	
	0	-1	
A	-1	2	

Fonte: Autor

Quando visualizamos a Figura 23, a interface tem quatro quadros, representando o nodo principal, na posição direita inferior, e sua vizinhança. Um nodo vizinho apresenta um valor central definido pelo algoritmo cadastrado e uma fórmula em sua área superior.

**Figura 23 - Detalhe de um nodo selecionado**



Fonte: Autor

A fórmula é uma representação detalhada (exemplo na Figura 20), que será exemplificada abaixo:

#### 4.2.1 Nodo diagonal

O nodo é o primeiro quadro da direita para esquerda. Ao analisar a fórmula “ $3 + (-2) = 1$ ”, o valor 3 é o valor do nodo em questão, somando com o valor -2, é o resultado do *mismatch*, pois a ligação entre a sequência A e B não apresentou valores iguais. Caso ocorresse ligação de valores iguais, a soma seria realizada pelo *match*.

#### 4.2.2 Nodo superior e esquerdo

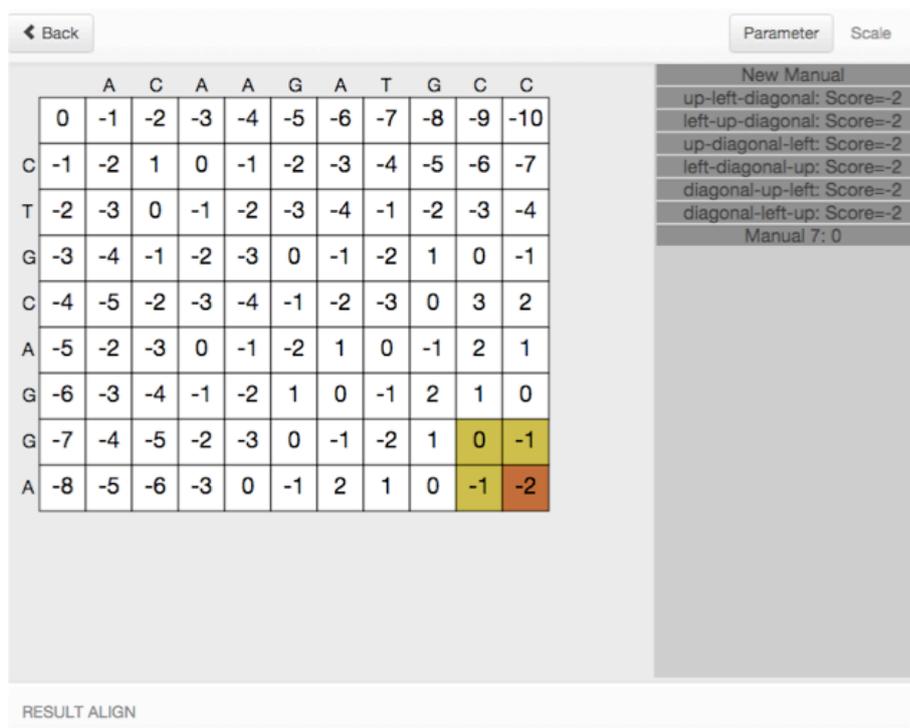
Esses nodos têm comportamento semelhante ao nodo diagonal, com a diferença de que o valor a ser somado ao valor do nodo trata-se do *gap*, ao visualizar a fórmula “ $2 + (-1) = 1$ ” e “ $6 + (-1) = 5$ ”.



### 4.2.3 Construção do alinhamento manual

Com base às informações anteriores será exibido um processo completo de um alinhamento global manualmente. Ao clicar em no campo “novo manual” o sistema inicia o processo de alinhamento como é possível visualizar na Figura 25.

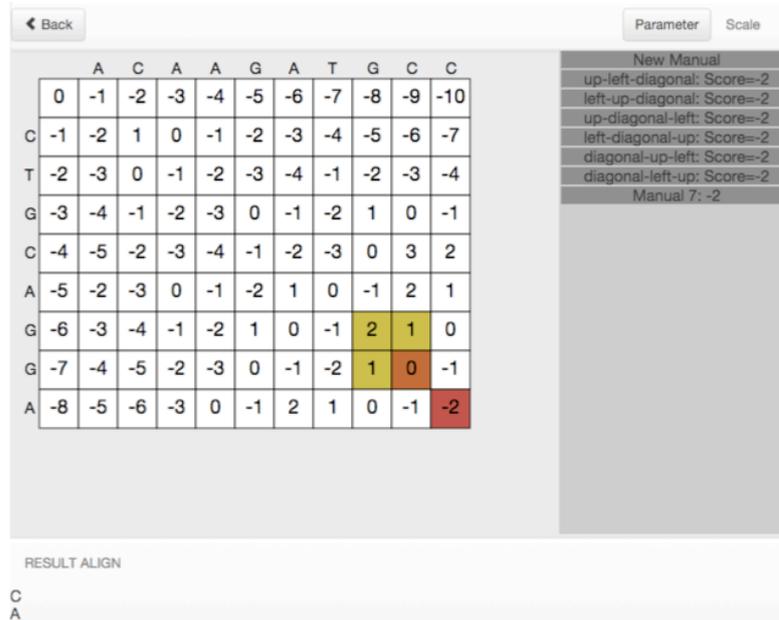
Figura 25 – Alinhamento manual (passo 1)



Fonte: Autor

Ao iniciar o processo de alinhamento manual, no caso utilizado no alinhamento global, o início já apresenta selecionado em laranja um nodo e traz neste caso três possibilidades de nodos para realizar a ligação da sequência apresentado em amarelo. Na Figura 26 foi selecionado o nodo com o valor zero, no caso seguindo o alinhamento e apresentando o primeiro resultado com o valor C e A.

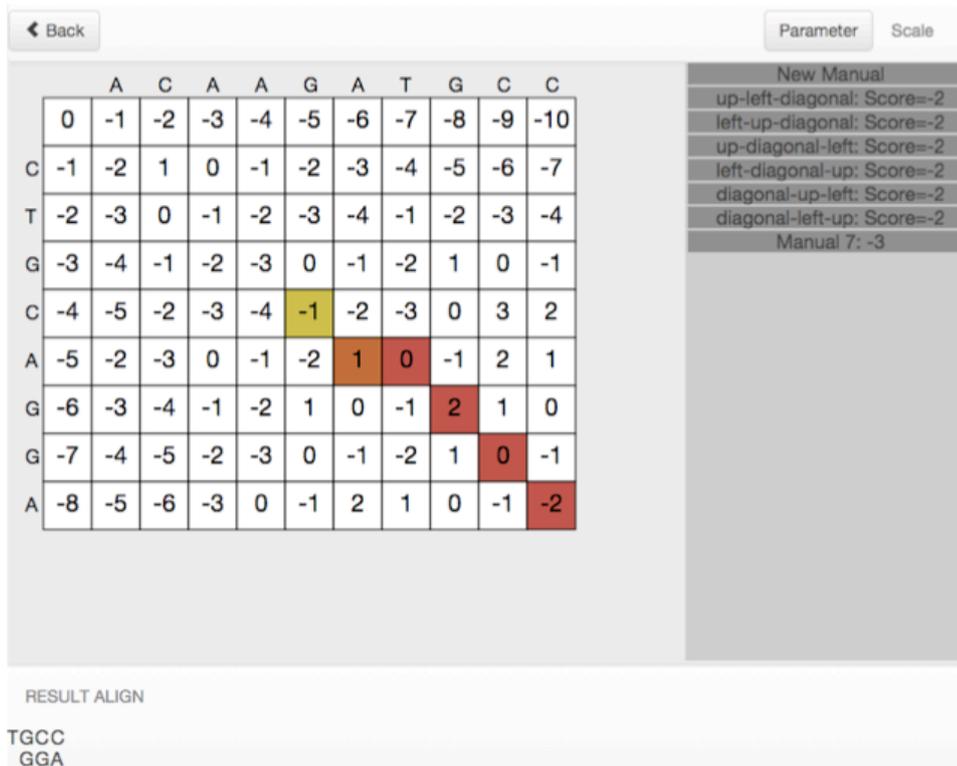
**Figura 26 - Alinhamento manual (passo 2)**



Fonte: Autor

Analizado a Figura 27, vemos que o processo apresenta um alinhamento ainda não concluído, mas com uma grande parte já apresentado o resultado e o score parcialmente.

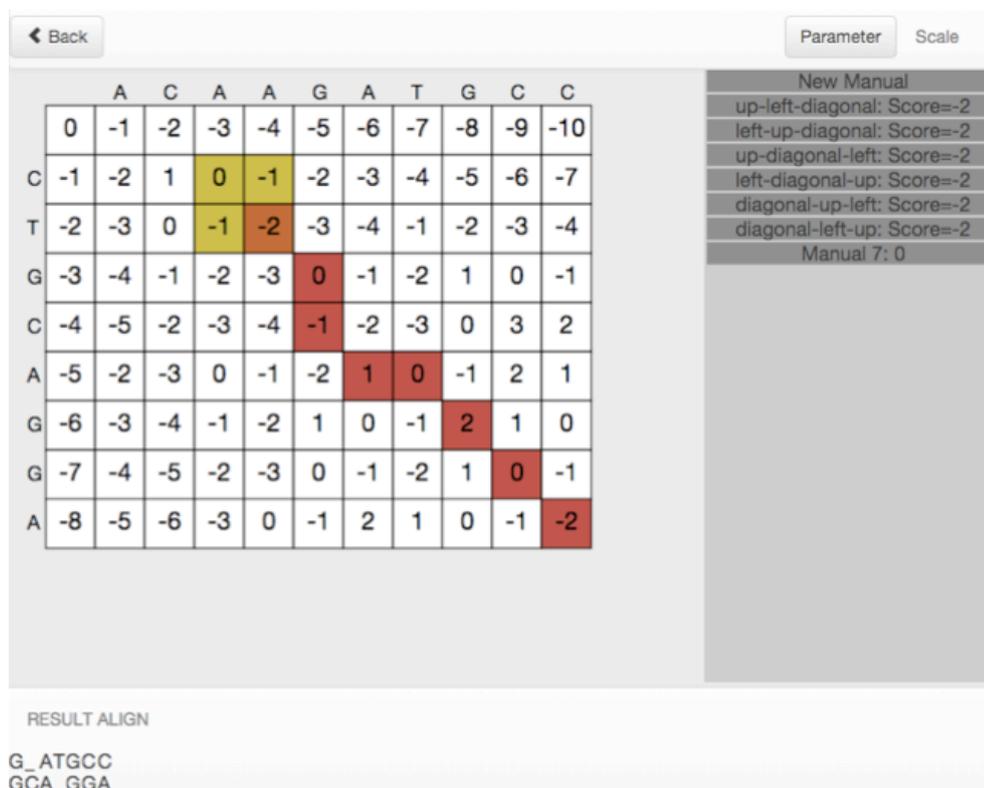
**Figura 27 - Alinhamento manual (passo 3)**



Fonte: Autor

Analisado a Figura 28, verificamos que o processo apresentou três possibilidades de ligação, para obter um melhor detalhamento da ligação basta clicar com o botão direito sobre o nodo, apresentado na Figura 29.

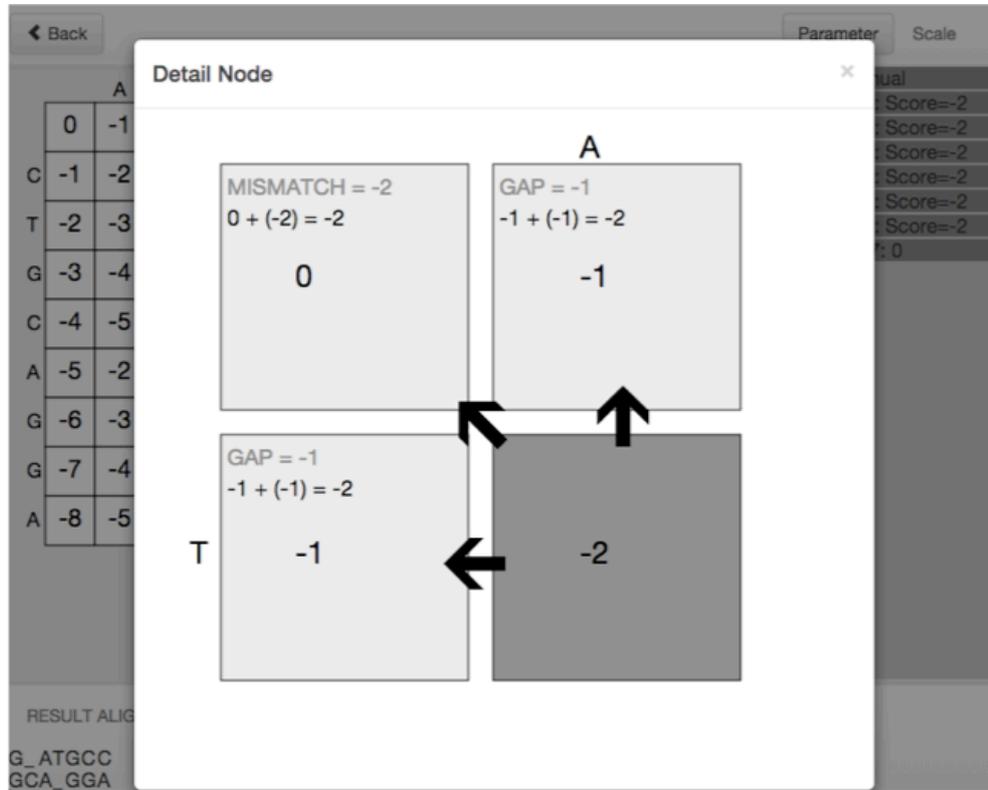
**Figura 28 - Alinhamento manual (passo 4)**



Fonte: Autor

Ao analisar a Figura 29, é possível notar as ligações aprestada anteriormente em amarelo, por setas e com todo o processo calculado, assim sendo possível exemplificar o processo. No uso de uma ferramenta com *BLAST*, a mesma apresenta somente o resultado o alinhamento, mas no caso do DNA Sequence é possível informar qual o melhor caminho a ser escolhido.

Figura 29 - Alinhamento manual detalhanto (passo 4)



Fonte: Autor

No próximo capítulo serão tratadas as estruturas de alinhamentos e interações entre a interface e a codificação. Serão descritos os principais processos de alinhamento demonstrados em capítulos anteriores.

## 5 ESTRUTURAS DE ALINHAMENTO DA DNA-SEQUENCE

Neste capítulo será descrito a estrutura e o conceito de alinhamento de sequência, identificando correlação entre os pares.

Segundo Larson, (2006) a bioinformática possui mais de 140 programas que compõem os pacotes de análise e comparação de sequência de proteínas e nucleotídeos que são encontrados em banco de dados como *EMBL*, *GenEMBL*, *PIR-Protein*, *Restriction Enzyme Database* e *GenBank*.

Segundo Baxevanis e Ouellette (2001), a ideia básica por trás de programação dinâmica vem da observação de subcaminhos provisórios que terminam em um ponto. Ao longo do caminho verdadeiro e ideal deve ser ele mesmo o caminho ideal que leva a esse ponto. Assim, o caminho ótimo pode ser encontrado, por extensão periódica de subcaminhos ideais.

Através do constante avanço da ciência e das tecnologias, é possível analisar que o ensino de Biologia e Ciências ainda permanecem, em muitos casos, limitados às aulas expositivas com mínima participação dos alunos. A utilização de outras modalidades didáticas é essencial, tais como: audiovisuais, ferramentas computacionais, etc. (LEPIENSKI; PINHO, 2014, p. 2).

Segundo Pevzner (2011, p.18), a bioinformática se tornou parte essencial da biologia moderna, principalmente quando se realiza pesquisas biológicas com as ferramentas de bioinformática modernas, como BLAST ou *Basic Local Alignment Search Tool*.

### 5.1 CÁLCULO DO GRÁFICO DE PONTOS

Segundo Lesk (2005), o gráfico de pontos é uma forma simples de visualização que fornece um panorama da similaridade entre duas sequências (par a par). O gráfico de pontos, presente na Figura 30, demonstra a identidade entre o nome abreviado (DOROTHYHODGKIN) e o nome completo (DOROTHYCROWFOOTHODGKIN) de uma famosa cristalógrafa de proteínas.

**Figura 30 - Exemplo de comparação**

	D	O	R	O	T	H	Y	C	R	O	W	F	O	O	T	H	O	D	G	K	I	N
D	D																					
O		O		O						O			O	O				O				
R			R							R												
O		O		O						O			O	O				O				
T					T															T		
H						H															H	
Y							Y															
H																						H
O		O		O						O			O	O								O
D	D																					D
G																						G
K																						K
I																						I
N																						N

Fonte: Adaptado de Lesk (2005)

## 5.2 CÁLCULO DO ALINHAMENTO

O cálculo de pares de alinhamento é feito em etapas. Primeiramente, é necessário criar a matriz ou gráficos de ponto com a ligação entre o elemento da sequência A (DOROTHYHODGKIN) e a sequência B (DOROTHYCROWFOOTHODGKIN), como no exemplo na Figura 31.

**Figura 31 - Disposição das sequencias na matriz**

	D	O	R	O	T	H	Y	C	R	O	W	F	O	O	T	H	O	D	G	K	I	N
D																						
O																						
R																						
O																						
T																						
H																						
Y																						
H																						
O																						
D																						
G																						
K																						
I																						
N																						

Fonte: Autor

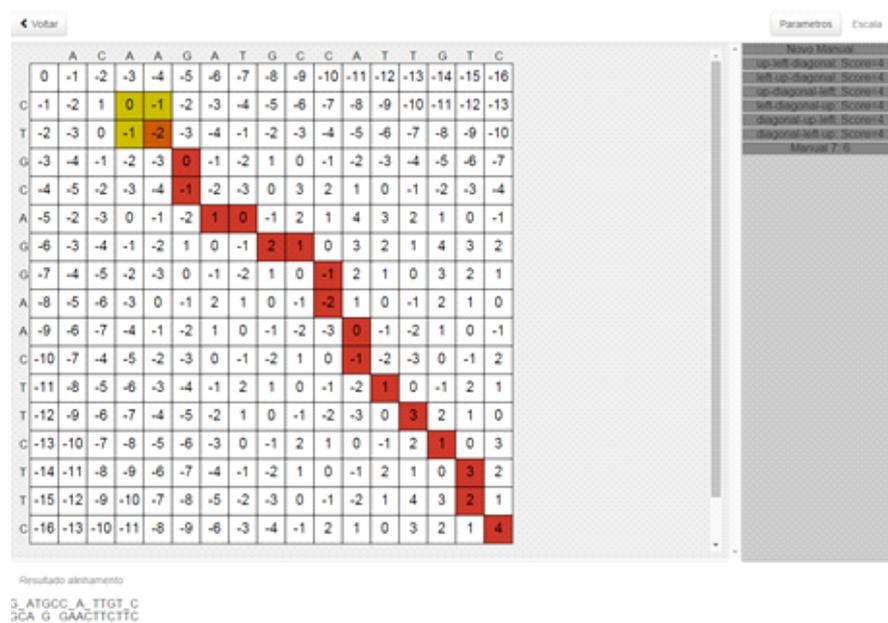
Ao realizar o estudo do alinhamento com base no uso de matriz de pontos, identifica-se que os confrontos entre valores apresentam resultados diferenciais perante aos valores não covalentes.

Considerando uma matriz no plano cartesiano no quarto quadrante, é representada por e com seus valores absolutos. Ao se calcular um elemento da matriz, leva-se em consideração os três vizinhos: os *gap* ( $x - 1, y$ ), ( $x, y - 1$ ) e o *match* ou *mismatch* ( $x - 1, y - 1$ ).

### 5.2.1 Cálculo do alinhamento global

O algoritmo de alinhamento global (*Needleman-Wunsch*) é resultado do cálculo da matriz de valores que os comportamentos resultam uma amostra dos pares alinhados de forma central. O que conhecemos como o algoritmo de *Needleman-Wunsch* é uma aplicação de uma estratégia de melhor caminho, chamado de programação dinâmica para o problema de encontrar o melhor alinhamento (NEEDLEMAN; WUNSCH, 1970).

Figura 32 - Alinhamento global



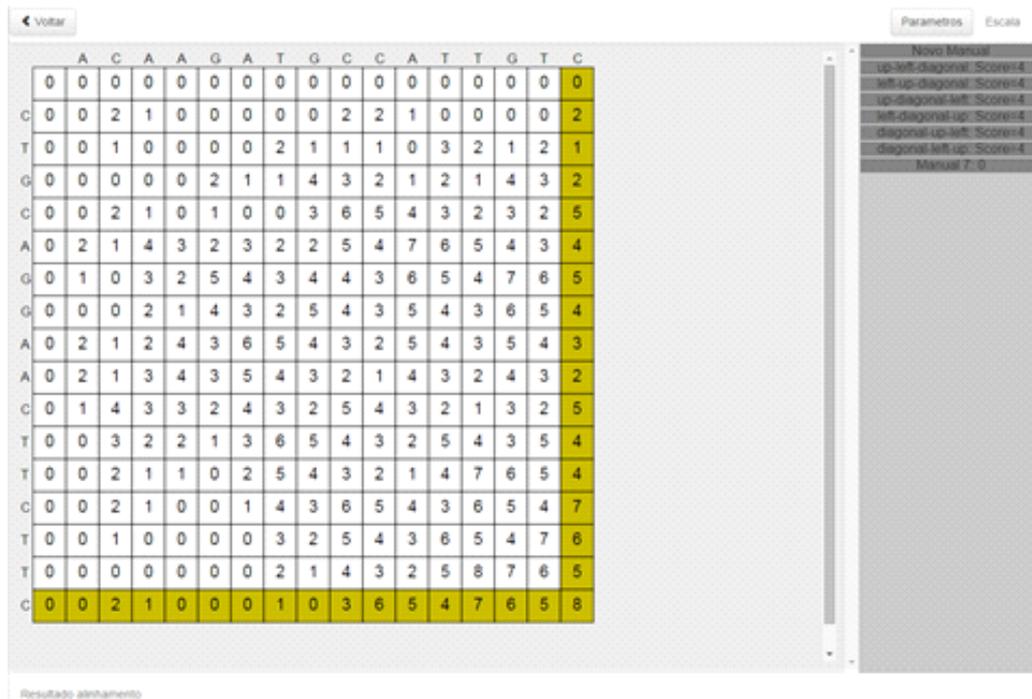
Fonte: Autor

No alinhamento global, representado na Figura 32, no exemplo, é realizado o método manual. Ao analisá-lo, tem seu início na extremidade inferior direito e seu final na posição superior esquerdo. No exemplo, nota-se que o valor em laranja apresenta três ligações em amarelo e isso significa que o algoritmo calculado da sua vizinhança apresenta o mesmo resultado, assim havendo a ligação entre todos os nodos.

### 5.2.2 Cálculo do alinhamento local

Outro alinhamento derivado do global é o alinhamento local (*Smith-Waterman*). Essa forma de alinhamento igualmente envolve a geração de uma matriz, mas localiza os pares de forma deslocada do centro (SETUBAL, J.; MEIDANIS J., 1997).

Figura 33 - Alinhamento local



Fonte: Autor

O alinhamento local representado na Figura 33 é realizado o método manual. Ao analisá-lo, percebe-se que ele não possui um início definido, diferente do alinhamento global. Assim, é necessário informar o primeiro nodo destacado em amarelo, após o processo se assemelha o método global.

Figura 34 - Exemplo de alinhamento concluído

	A	C	A	A	G	A	T	G	C	C	A	T	T	G	T	C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	2	1	0	0	0	0	2	2	1	0	0	0	0	2
T	0	0	1	0	0	0	0	2	1	1	1	0	3	2	1	2
G	0	0	0	0	0	2	1	1	4	3	2	1	2	1	4	3
C	0	0	2	1	0	1	0	0	3	6	5	4	3	2	3	5
A	0	2	1	4	3	2	3	2	2	5	4	7	6	5	4	3
G	0	1	0	3	2	5	4	3	4	4	3	6	5	4	7	6
G	0	0	0	2	1	4	3	2	5	4	3	5	4	3	6	5
A	0	2	1	2	4	3	6	5	4	3	2	5	4	3	5	4
A	0	2	1	3	4	3	5	4	3	2	1	4	3	2	4	3
C	0	1	4	3	3	2	4	3	2	5	4	3	2	1	3	5
T	0	0	3	2	2	1	3	6	5	4	3	2	5	4	3	5
T	0	0	2	1	1	0	2	5	4	3	2	1	4	7	6	5
C	0	0	2	1	0	0	1	4	3	6	5	4	3	6	5	4
T	0	0	1	0	0	0	0	3	2	5	4	3	6	5	4	7
T	0	0	0	0	0	0	0	2	1	4	3	2	5	8	7	6
C	0	0	2	1	0	0	0	1	0	3	6	5	4	7	6	5

RESULT ALIGN  
 \_\_ACA\_\_AGA\_\_\_\_TGC  
 CTGCAGGA\_ ACTTCTT C

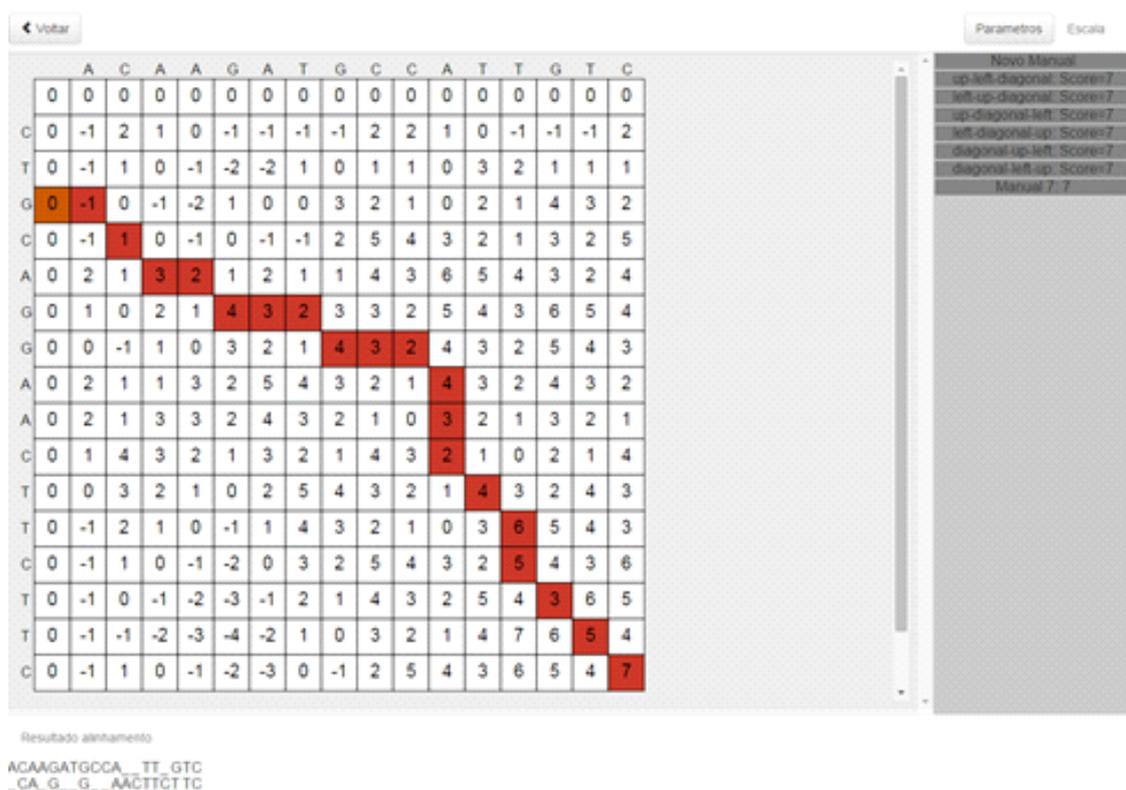
Fonte: Autor

No exemplo da Figura 34, é possível visualizar o processo concluído. A diferença presente no processo está no começo do dele, pois não é obrigatório o seu início na extremidade da matriz e os nodos não apresentam valores negativos. Caso o cálculo apresente valor negativo, é trocado para zero.

### 5.2.3 Cálculo do alinhamento semi-global

Este alinhamento é uma derivação do alinhamento global. A construção da matriz é igual ao global com definição de *gap*, *mismatch* e *match*. A diferença está na identificação do início do alinhamento igual ao alinhamento local e o fim do alinhamento, presente em qualquer extremidade, tanto na vertical ou horizontal da matriz. A Figura 35 apresenta o alinhamento semi-global.

Figura 35 - Alinhamento semi-global



Fonte: Autor

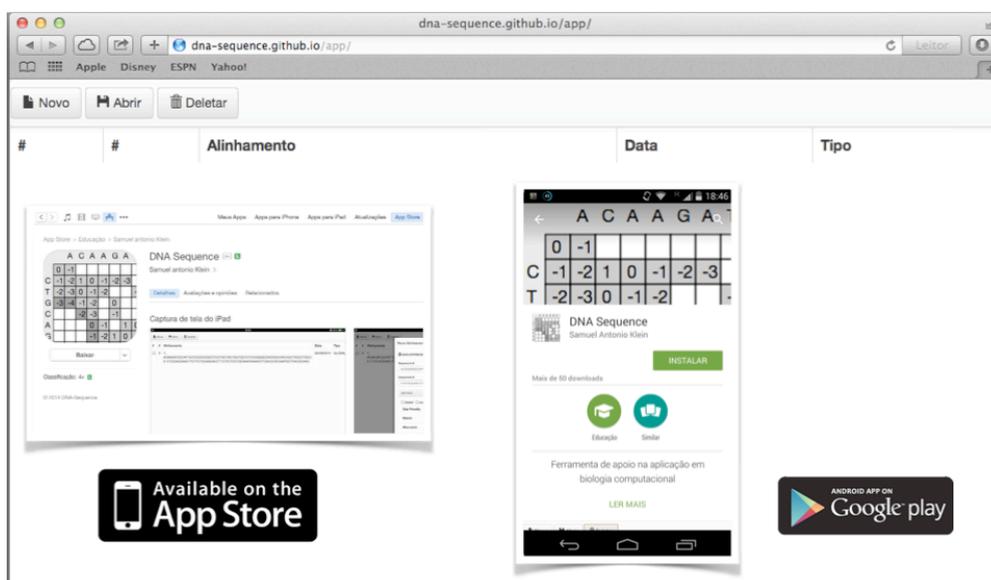
O alinhamento semi-global, na Figura 35, é realizado o método manual. Ao analisá-lo, vê-se o início destacado pelo maior valor de suas extremidades. Caso haja mais de um igual, é necessário escolher o mesmo destacado em amarelo. Após o processo se assemelha ao método global, com a diferença de que o final do alinhamento pode ser concluído em qualquer extremidade da esquerda ou superior.

O estudo para entender os comportamentos do resultado de um alinhamento, muitas vezes, torna-se complexo para um aluno de biologia, pois não compreende as essências entre os dados disponibilizados e os resultados apresentados.

## 6 DISTRIBUIÇÃO DO SOFTWARE

A distribuição da ferramenta está disponível diretamente no browser pelo link <http://dna-sequence.github.io/app>, além disso é possível encontrar a versão mobile na loja da Google à Google Play ([https://play.google.com/store/apps/details?id=br.com.samuelklein.dna.phonegap.DNA\\_Sequence](https://play.google.com/store/apps/details?id=br.com.samuelklein.dna.phonegap.DNA_Sequence)) e na loja na Apple à AppStore (<https://itunes.apple.com/us/app/dna-sequence/id923854396?ls=1&mt=8>). A Figura 36 apresenta a disponibilidade em dispositivo movel é uma forma excelente de distribuição.

**Figura 36 – Distribuição**



Fonte: Autor

De acordo com Faleiro, Andrade e Junior (2011) afirmam que o grande avanço obtido até os dias de hoje, a rapidez com que surgem novas técnicas da biologia molecular e o gigantesco volume de dados e informações produzidos pelos projetos nessa área exigem que a bioinformática esteja em constante evolução, para um aperfeiçoamento de ferramentas de bioinformática para o desenvolvimento.

## CONCLUSÃO

Com a elaboração deste trabalho, foi possível analisar o estudo e o processo de alinhamento e sequenciamento de DNA, contemplando a maioria dos conceitos apresentados didaticamente em livros e artigos.

Através do referencial teórico, foi possível elaborar uma ferramenta de alinhamento, contendo processos, regras e métodos para um entendimento do passo a passo em comparação de sequências.

No processo de desenvolvimento foi estudado diversas maneiras de implementar as estruturas de alinhamentos. Com o conhecimento já adquirido em programação *Java*, foi idealizado a utilização de interface *swing* com programa somente *Java*.

No desenvolvimento se verificou a necessidade de uma ferramenta multiplataforma, e, dessa forma, decidiu-se por uma implementação voltada à *web*, com páginas estáticas e comunicação em *WebService REST*.

No entanto, o uso de *WebService* deixa o processo de alinhamento lento, dependendo inteiramente da velocidade do servidor disponível. Com isso, surgiu a ideia de realizar todo o processo de alinhamento na máquina do cliente, mas o processo permaneceu implementado em *Java*.

O estudo progrediu para uma ideia de um conversor de *Java* para *JavaScript*. Esta tecnologia já utilizada pelo GWT, com algumas diferenças e algumas dificuldades, solucionadas foi adotada.

Assim, buscou-se facilitar o desenvolvimento em *Java* dos processos de alinhamento, sem a necessidade de um servidor de contexto *web* em *Java*, mas apenas um servidor *web* simples.

Entre os pontos fortes do uso da ferramenta, estão a independência em *software* adicionais e a sua disponibilidade gratuita e com seu fonte aberto para colaboração de pessoas interessadas em aprimorar os processos ou implementar novas definições.

Todos os fontes e ferramentas utilizadas para o desenvolvimento foram disponibilizados em repositórios colaborativos e sem o uso de ferramentas proprietárias para o desenvolvimento do mesmo.

A disponibilização da ferramenta com o uso do *GitHub* teve um propósito importante, pois o controle de infraestrutura da página deixa de ser um trabalho pessoal. Dessa forma, é possível ter um foco na ferramenta.

Além da disponibilidade da ferramenta em uma página *web*, foi elaborado teste com o uso de dispositivos móveis. Os testes foram realizados em plataforma *Android* e *IOS*, com o uso do *iPad*. Após as realizações de testes, nota-se que o *software* teria que passar por um processo de compatibilidade com o meio de dispositivos móveis, como as interações do *mouse versus* tela sensível ao toque. Mesmo com a dificuldade, foi possível executar a ferramenta em um dispositivo móvel.

Os alinhamentos apresentados no *software*, utilizando teste unitário, tiveram sucesso em seus resultados, não apresentando erro de cálculos ou algoritmos de acordo com o método utilizado.

De acordo com cada método, o *software* apresenta técnicas diferentes de apresentação. Dessa forma, foi necessário elaborar estruturas específicas de cada método, para que fosse possível atingir o seu objetivo, auxiliando o entendimento do processo de alinhamento.

Foi realizada a avaliação sobre a sua usabilidade, utilidade e satisfação, conforme pode ser conferido no questionário em anexo no Apêndice D. Através dos resultados, foi possível ter uma visão diferenciada sobre o *software* e as outras funcionalidades importantes no uso da ferramenta. Muitas dessas observações e comentários presentes nas avaliações pode ser implementado em versão futuras. O uso da técnica de avaliação apresentada baseada Nielsen (2001), não necessariamente seria a melhor forma de avaliação no caso na ferramenta apresentada.

O *software* apresentar três técnicas de alinhamento, abrindo, com isso, uma possibilidade de novas implementações e novos meios de apoio ao aprendizado de bioinformática.

Nesse ano de estudo e elaboração da **ferramenta de apoio na aplicação de algoritmos de alinhamento em biologia computacional**, os objetivos foram sendo concluídos conforme o software foi desenvolvido. A ferramenta evoluiu com novos métodos e melhor usabilidade e com o uso de tutorias sendo uma proposta excelente para elaboração de trabalhos futuros.

Como trabalhos futuros se sugere a implantação de mais métodos de alinhamento, implementação de uso de tabela de pontuação para o alinhamento de proteínas e a consideração de métodos de alinhamento múltiplo de sequências com mais de duas sequências por análise. Com futuro seria necessário a implementação e aplicação de ferramenta de avaliação levando em considerações fatores do uso da ferramenta no entendimento de programação dinâmica e no aprimoramento do uso didático no estudo de alinhamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTS, Bruce et al. **Biologia molecular da célula**. 5 ed. Porto Alegre: Artmed, 2008.

ALPHEY, Luke. **DNA Sequencing: From Experimental Methods to Bioinformatics**. New York: Springer, 1997.

BAXEVANIS, Andreas D.; OUELLETTE, B.F. Francis. **Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins**. 2 ed. New York US: WILEY, 2001.

BORODOVSKY, Mark; EKISHEVA, Svetlana. **Problems and solutions in biological sequence analysis**. Cambridge: Cambridge, 2006.

CHAITIN, Gregory. **Meta Math: The quest for omega**. New York: Vintage, 2006.

CHANDAR, Nalini; VISELLI, Susan. **Biologia celular e molecular ilustrada**. Porto Alegre: Artmed, 2011.

CHEN, Peter. **Gerenciando banco de dados: a abordagem entidade-relacionamento para projeto lógico**. São Paulo, SP: McGraw-Hill, 1990.

CLAVERIE, Jean-Michel; NOTREDAME, Cedric. **Bioinformatics for dummies**. Hoboken: Wiley Publishing, 2003.

CONOLEY, Chris. **Chemistry**. 3ª Ed. London: Harper-Collins, 2008.

Creationwiki. **Gel electrophoresis.PNG**. Disponível em <[http://creationwiki.org/pool/images/thumb/5/5e/Gel\\_electrophoresis.PNG/474px-Gel\\_electrophoresis.PNG](http://creationwiki.org/pool/images/thumb/5/5e/Gel_electrophoresis.PNG/474px-Gel_electrophoresis.PNG)>. Acesso em 10 out. 2014.

CRICK, Francis. **Central Dogma of Molecular Biology**. Nature: vol. 227. 8 de agosto, 1970.

FALEIRO, Fabio Gelape; Andrade, Solange R. M.; JUNIOR, Fabio B. dos R. **BIOTECNOLOGIA Estado da arte e aplicações na agropecuária**. Embrapa Cerrados, 2011.

FOWLER, Martin. **UML essencial**. 2 ed. Porto Alegre: Bookman, 2005.

GEIGER, Dan. Introduction to Algorithms in Computational Biology: Lecture 1. Disponível em: <<http://www.cs.technion.ac.il/~dang/courseCB/xlecture01.pdf>> Acesso em 10 de jun. de 2014.

GIBAS, Cynthia; JAMBECK, Per. **Desenvolvendo Bioinformática**, Rio de Janeiro: Campus, 2001.

GRIFFITHS, Anthony J. F.; WESSLER, Susan R.; LEWONTIN, Richard C.; CARROLL, Sean B. **Introdução à Genética**. 9ª ed. Rio de Janeiro: Guanabara Koogan, 2009.

HANSON, Robert; TACY, Adam. **GWT in Action: Easy Ajax with the Google Web Toolkit**. Shelter Island, NY: Manning, 2007.

HARTL, Daniel L. **Essential genetics: A genomics perspective**. Sudbury: Jones and Bartlett, 2011.

\_\_\_\_\_. **Princípios de Genética de População**. São Paulo: Funpec Editora, 2008.

JONES, Neil C.; PEVZNER, Pavel A. **An introduction to bioinformatic algorithms**. 8 ed. Cambridge: The MIT Press, 2004.

JUNQUEIRA, Dennis M.; BRAUN, Rodrigo L.; VERLI, Hugo. Alinhamentos. In.: VERLI, Hugo (org.). **Bioinformática: Da biologia à flexibilidade molecular**. Porto Alegre, 2014.

JUNQUEIRA, Luis C.; CARNEIRO, José. **Biologia celular e molecular**. 8 ed. Rio de Janeiro: Guanabara-Koogan, 2011.

KARP, Gerald. **Cell and Molecular Biology: Concepts and Experiments**. 6 ed. US: John Wiley & Sons, 2009.

KORF, Ian; YANDELL, Mark; BEDELL, Joseph. **Blast**, Cambridge: O'Reilly, 2003.

LARSON, Richard S. **Bioinformatics and Drug Discovery**. New Jersey US: HUMANA PRESS, 2006.

LEPIENSKI, Luis Marcos; PINHO, Kátia E. Prus. **Recursos didáticos no ensino de biologia e ciências**. Disponível em <<http://www.diadiaeducacao.pr.gov.br/portals/pde/arquivos/400-2.pdf>>. Acesso em 19 abr. 2013.

LESK, Arthur M. **Introdução à bioinformática**. 2ª ed. Porto Alegre: ARTMED, 2005.

LOELIGER, Jon; MCCULLOUGH, Matthew. **Version Control with Git: Powerful Tools and Techniques for Collaborative Software**. 2 ed. Sebastopol, CA: O'Reilly Media, Inc, 2012.

MARKELL, Scott; DARRYL, León;. **Sequence Analysis in a Nutshell**. Sebastopol, CA: O'Reilly Media, Inc, 2003.

MAXAM, A. M.; GILBERT W. **A new method for sequencing DNA**. Proc. Natl. Acad. Sci. v. 74. n. 2, p. 560-564, 1977.

NEEDLEMAN, Saul B.; WUNSCH, Christian. **A general method applicable to the search for similarities in the amino acid sequence of two proteins**. J. Mol. Biol. 48, p. 443–453, 1970.

NIELSEN, J.; MACK, Robert. **Usability Inspection Methods**. New York: John Wiley & Sons, 2001.

PEVZNER, Pavel. **Computational molecular biology: An algorithmic approach**. Cambridge: The MIT Press, 2000.

\_\_\_\_\_. Educating biologists in the 21st century: Bioinformatics scientists versus bioinformatics technicians. In.:MANDOIU, I. I.; ZELIKOVSKY, A. (orgs.).**Bioinformatics algorithms: Techniques and applications**. Hoboken: John Wiley & Sons, 2008. p.1 – 6.

PEVZNER, Pavel; SHAMIR, Ron. **Bioinformatics for Biologists**. Cambridge UK: UNIVERSITY PRESS, 2011.

RAMSDEN, Jeremy. **Bioinformatics: An Introduction**. Springer Science & Business Media, 2009.

RINGO, John. **Genética Básica**. Rio de Janeiro: Guanabara-Koogan, 2005.

SANGER, F; COULSON, A. R. **A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase**. J. Mol. Biol. v. 94. n. 3, p. 441-448, 1975.

SAYED Ali Sheikh Md; XIA, Ke; YANG, Tian-Lun; PENG, Jun. Circulating microRNAs: A potential role in diagnosis and prognosis of acute myocardial infarction. **Disease Makers**, v. 35, n. 5, p. 561-566, 2013.

SETUBAL, Carlos; MEIDANIS, João. **Introduction to Computational Molecular Biology**. Boston: PWS, 1997.

SHARMA, Kal Renganathan. **Bioinformatics: Sequence alignment and markov models**. New York: McGraw-Hill, 2009.

SOMMERVILLE, Ian. **Software engineering**. Edinburg Gate: Addison-Wesley, 2001.

WATERMAN, Michael S. **Introduction to Computational Biology: Maps, sequences and genomes**, Boca Raton: G Chapman & Hall, 2000.

WATSON, James D.; BAKER, Tania A; BELL, Stephen P.; GANN, Alexander; LEVINE, Michael; LOSICK, Richard. **Biologia Molecular do Gene**, Porto Alegre: Artmed, 2004.

WikimediaCommons. **DNA\_chemical\_structure\_pt.svg**. Disponível em <[http://upload.wikimedia.org/wikipedia/commons/thumb/5/52/DNA\\_chemical\\_structure\\_pt.svg/500px-DNA\\_chemical\\_structure\\_pt.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/5/52/DNA_chemical_structure_pt.svg/500px-DNA_chemical_structure_pt.svg.png)>. Acesso em 16 jun. 2014.

\_\_\_\_\_. **Hemoglobin.jpg**. Disponível em <<http://upload.wikimedia.org/wikipedia/commons/1/19/Hemoglobin.jpg>>. Acesso em 16 jun. 2014.

\_\_\_\_\_. **Pcr\_machine.jpg**. Disponível em <[http://commons.wikimedia.org/wiki/File:Pcr\\_machine.jpg](http://commons.wikimedia.org/wiki/File:Pcr_machine.jpg)>. Acesso em 16 jun. 2014.

YE, Yuzhen; TANG, Haixu. Dynamic programming algorithms for biological sequence and structure comparison. In.: MANDOIU, I. I.; ZELIKOVSKY, A. (orgs.). **Bioinformatics algorithms: Techniques and applications**. Hoboken: John Wiley & Sons, 2008. p.9-28.

## APÊNDICE A – ESTRUTURA JSON DO ALINHAMENTO GLOBAL DAS SEQUENCIAS (“ABCDE”, “ABCEF”)

### Entrada para geração da matrix

```
{
  "typeElement": "",
  "sequenceA": "ABCDE",
  "sequenceB": "ABCEF",
  "methodSequencing": "GLOBAL",
  "gap": "-1",
  "match": "2",
  "misMatch": "-2",
  "arrayMisMatch": ["A-B", -3],
  "id": 6
}
```

### Entrada exemplo de alinhamento global

```
{
  "idMatrix": 6,
  "connecteds": [
    "N",
    "W",
    "NW"
  ]
}
```

### Retorno do alinhamento global

```
{
  "nodes": [
    {
      "value": 2,
      "connected": [
        "N",
        "W",
        "NW"
      ],
      "candidate": true,
      "x": 5,
      "y": 5,
      "charSeqA": "E",
      "charSeqB": "F"
    },
    {
      "value": 3,
      "connected": [
        "N",
        "W",
        "NW"
      ],
      "candidate": true,
      "x": 5,
      "y": 4,
      "charSeqA": "E",
      "charSeqB": "C"
    },
    {
      "value": 4,
      "connected": [
        "W"
      ],
      "candidate": true,
      "x": 5,
      "y": 3,
      "charSeqA": "E",
      "charSeqB": "D"
    }
  ]
}
```

```

    "value":5,
    "connected":[
      "NW"
    ],
    "candidate":true,
    "x":4,
    "y":3,
    "charSeqA":"D",
    "charSeqB":"D"
  },
  {
    "value":3,
    "connected":[
      "W"
    ],
    "candidate":true,
    "x":3,
    "y":2,
    "charSeqA":"C",
    "charSeqB":"B"
  },
  {
    "value":4,
    "connected":[
      "NW"
    ],
    "candidate":true,
    "x":2,
    "y":2,
    "charSeqA":"B",
    "charSeqB":"B"
  },
  {
    "value":2,
    "connected":[
      "NW"
    ],
    "candidate":true,
    "x":1,
    "y":1,
    "charSeqA":"A",
    "charSeqB":"A"
  },
  {
    "value":0,
    "connected":[
    ],
    "candidate":true,
    "x":0,
    "y":0
  }
],
"resultSequenceA":"ABCDE__",
"resultSequenceB":"AB_D_CF"
}

```

## APÊNDICE B – ESTRUTURA DAS CLASSES DA FERRAMENTA DNA-SEQUENCE

### Classe que representa um nodo de uma matriz

```

/**
 * @file dna-sequence-process
 * @namespacedna
 * @licence GNU GPL v3
 * @copyright Copyright (c) 2014.
 * Created by samuel on 17/06/14.
 */

/**
 * @global
 * @constructor
 */
window.dna.Node = function () {

};

/**
 *
 * @type {{value: {int}, connected: Array, candidate: boolean, x: {int}, y: {int}, charSeqA:
 {String}, charSeqB: {String}}}
 */
window.dna.Node.prototype = {
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {int}
   */
  value: null,
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {Array.<Connected>}
   */
  connected: [],
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {boolean}
   */
  candidate: false,
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {int}
   */
  x: null,
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {int}
   */
  y: null,
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {String}
   */
  charSeqA: null,
  /**
   * @global
   * @memberOf Node
   * @instance
   * @type {String}
   */

```

```
charSeqB: null
};
```

## Classe que representa uma matriz

```
/**
 * @file dna-sequence-process
 * @namespacedna
 * @licence GNU GPL v3
 * @copyright Copyright (c) 2014.
 * Created by samuel on 17/06/14.
 */

/**
 * @global
 * @constructor
 */
window.dna.Matrix = function () {

};

window.dna.Matrix.prototype = {
  /**
   * @memberOf Matrix
   * @instance
   * @type {Array.<Node>}
   */
  nodes: [],
  /**
   * @memberOf Matrix
   * @instance
   * @type {Array.<string>}
   */
  sequenceA: [],
  /**
   * @memberOf Matrix
   * @instance
   * @type {Array.<string>}
   */
  sequenceB: []
};
```

## Classe que representa o detalhamento do Nodo quando clicado em um nodo

```
/**
 * @file dna-sequence-process
 * @namespacedna
 * @licence GNU GPL v3
 * @copyright Copyright (c) 2014.
 * Created by samuel on 23/08/14.
 */

/**
 * @global
 * @constructor
 */
window.dna.NodeDetail = function () {

};

window.dna.NodeDetail.prototype = {
  /**
   * @global
   * @memberOfNodeDetail
   * @instance
   * @type {Node}
   */
  node: null,
  /**
   * @global
   * @memberOfNodeDetail
   * @instance
   */
}
```

```
        * @type {Node}
    */
    nodeW: null,
    /**
     * @global
     * @memberOfNodeDetail
     * @instance
     * @type {Node}
     */
    nodeNW: null,
    /**
     * @global
     * @memberOfNodeDetail
     * @instance
     * @type {Node}
     */
    nodeN: null,
    /**
     * @global
     * @memberOfNodeDetail
     * @instance
     * @type {String}
     */
    nodeCalcW: null,
    /**
     * @global
     * @memberOfNodeDetail
     * @instance
     * @type {String}
     */
    nodeCalcNW: null,
    /**
     * @global
     * @memberOfNodeDetail
     * @instance
     * @type {String}
     */
    nodeCalcN: null
};
```

## APÊNDICE C – ARQUIVO DE PARÂMETRO INTERNACIONALIZAÇÃO

### Arquivo de parâmetro Português

```
{
  "menu": {
    "new": "Novo",
    "open": "Abrir",
    "delete": "Deletar"
  },
  "headAlign": {
    "Align": "Alinhamento",
    "Date": "Data",
    "Type": "Tipo"
  },
  "pageNew": {
    "NewAlign": "Novo Alinhamento",
    "NUCLEOTIDE": "NUCLEOTIDEOS",
    "PROTEIN": "PROTEINA",
    "SequenceA": "Sequência A",
    "SequenceB": "Sequência B",
    "METHOD": "METODO",
    "Global": "Global",
    "Local": "Local",
    "SemiGlobal" : "Semi-global",
    "Gap": "Gap Penalty",
    "Match": "Match",
    "Mismatch": "Mismatch",
    "Close": "Fechar",
    "Save": "Salvar"
  },
  "openAlign": {
    "Back": "Voltar",
    "SCALE": "Escala",
    "SCORE": "Score",
    "NEW_MANUAL": "Novo Manual",
    "ResultAlign": "Resultado alinhamento",
    "DetailNode" : "Detalhe Elemento",
    "Close": "Fechar",
    "Save": "Salvar",
    "ShowAllPaths" : "Mostrar todos os caminhos",
    "PARAMETER" : "Parâmetros"
  }
}
```

### Arquivo de parâmetro Inglês

```
{
  "menu": {
    "new": "New",
    "open": "Open",
    "delete": "Delete"
  },
  "headAlign": {
    "Align": "Align",
    "Date": "Date",
    "Type": "Type"
  },
  "pageNew": {
    "NewAlign": "New Align",
    "NUCLEOTIDE": "NUCLEOTIDE",
    "PROTEIN": "PROTEIN",
    "SequenceA": "Sequence A",
    "SequenceB": "Sequence B",
    "METHOD": "METHOD",
    "Global": "Global",
    "Local": "Local",
    "SemiGlobal" : "Semi-global",
    "Gap": "Gap Penalty",
    "Match": "Match",
    "Mismatch": "Mismatch",
    "Close": "Close",
    "Save": "Save"
  }
}
```

```
},
"openAligh": {
  "Back": "Back",
  "SCALE": "Scale",
  "SCORE": "Score",
  "NEW_MANUAL": "New Manual",
  "ResultAlign": "RESULT ALIGN",
  "DetailNode" : "Detail Node",
  "Close": "Close",
  "Save": "Save",
  "ShowAllPaths" : "Show all paths",
  "PARAMETER" : "Parameter"
}
}
```

## APÊNDICE D – AVALIAÇÃO DA FERRAMENTA DE APOIO AO ENSINO DE TIPOS DE ALINHAMENTO EM BIOINFORMÁTICA

### 1) Avaliação de Usabilidade:

1. Eu gostaria de utilizar esta ferramenta frequentemente na disciplina de bioinformática.

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

2. Achei a ferramenta desnecessariamente complexa (sendo a opção 1 para pouco complexa e a opção 5 para muito complexa).

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

3. Achei a ferramenta fácil de usar.

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

4. Imagino que precisaria de suporte técnico para poder usar esta ferramenta (sendo a opção 1 para baixa necessidade de suporte e a opção 5 para alta necessidade de suporte).

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

5. Achei que as diversas funções desta ferramenta foram bem integradas.

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

6. Penso que meus alunos de bioinformática aprenderiam a usar esta ferramenta rapidamente (sendo a opção 1 para a minoria da população e a opção 5 para a maioria da população).

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

7. Eu achei que a ferramenta possui limitações que iriam restringir a usabilidade.

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

### 2) Percepção de Utilidade:

1. A ferramenta tem potencial para dar auxílio ao ensino de bioinformática.

Discordo plenamente	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> </tr> </table>						Concordo plenamente
	1      2      3      4      5						

2. A ferramenta oferece motivação e engajamento durante o aprendizado (sendo a opção 1 para pouco motivante e engajante e a opção 5 para altamente motivante e engajante).

Discordo plenamente	<input type="checkbox"/>	Concordo plenamente				
	1	2	3	4	5	

### 3) Avaliação da Satisfação:

1. Qual foi a sua primeira impressão ao utilizar a ferramenta?