FACULDADES INTEGRADAS DE CARATINGA

FACULDADE DE CIÊNCIA DA COMPUTAÇÃO

ESTUDO COMPARATIVO DE ALGORITMOS PARA CASAMENTO DE PADRÕES EM FILAMENTOS DE ÁCIDOS NUCLÉICOS

ALISON TORRES RIBEIRO

CARATINGA

2010

Alison Torres Ribeiro

ESTUDO COMPARATIVO DE ALGORITMOS PARA CASAMENTO DE PADRÕES EM FILAMENTOS DE ÁCIDOS NUCLÉICOS

Monografia apresentada ao Curso de Ciência da Computação das Faculdades Integradas de Caratinga como requisito parcial para obtenção do título de Bacharel em Ciência da Computação orientado pelo Professor MSc. Filipe Costa Fernandes.

Caratinga
2010

Alison Torres Ribeiro

ESTUDO COMPARATIVO DE ALGORITMOS PARA CASAMENTO DE PADRÕESEM FILAMENTOS DE ÁCIDOS NUCLÉICOS

Monografia submetida à Comissão examinadora designada pelo Curso de Graduação em Ciência da Computação como requisito para obtenção do grau de Bacharel.

_____

Prof. MSc Filipe Costa Fernandes (Orientador)
Faculdades Integradas de Caratinga

_____

Prof. MSc Fabrícia Pires Souza Tiola
Faculdades Integradas de Caratinga

_____

Prof. MSc Paulo Eustáquio dos Santos
Faculdades Integradas de Caratinga

Caratinga, 10/12/2010

AGRADECIMENTOS

Agradeço a Deus por tudo que aconteceu na minha vida, além de tudo por estar vivo, agradeço a meus pais Ana e Nélio e a meus irmãos Túlio e Fernando, a minha cunhada Luana e a meu querido sobrinho Daniel um menino maravilhoso, sem o apoio deles não teria chegado até aqui.

Agradeço a meu orientador Filipe Costa Fernandes por ter me ajudado a concluir este trabalho, e também ao Professor Paulo Eustáquio outra pessoa que não me deixou mudar de idéia para fazer outro trabalho, e disse algo importante, que ao invés de olhar para esta mesa inteira e fazer o trabalho olhe apenas para esta manchinha da mesa e comece a trabalhar que você já fará algo bom.

“Não se constrói um destino apenas com palavras, mas com sonho, suor e sangue”

Alison Torres Ribeiro

RESUMO

Nos filamentos de ácidos nucléicos existem partes que são responsáveis por qualquer das funções existentes em seres vivos, e a partir destas partas pode-se entender melhor as formas de vida e sua base de funcionamento, pois elas são conservadas na evolução das espécies (Lemos et al.,2003).

Atualmente a maioria das doenças é provocada por vários tipos de vírus que possuem todas as suas funções encontradas em seu material genético e entendendo melhor o seu DNA pode-se prever a resposta para a cura destas doenças.

Com a elaboração dos algoritmos Força Bruta, Knuth Morris e Pratt, *Iterated Local Search* e heurística de Boyer Moore, estes algoritmos farão o casamento de padrões em filamentos de ácidos nucléicos indicando a quantidade de padrões encontrados, a posição dos padrões e o tempo gasto para a pesquisa.

Este trabalho tem como objetivo a apresentação destes algoritmos utilizados para busca de padrões em filamentos de DNA com o intuito de se prever qual o algoritmo encontrará o maior número de padrões em menor tempo possível.

Palavras-chave: Ácidos nucléicos, Força Bruta, Knuth Morris e Pratt, *Iterated Local Search*, Boyer Moore, DNA, RNA, Heurística.

ABSTRACT

In the nucleic acids filaments parts that are responsible for any of the existent functions in alive beings exist, and starting from these you leave can understand each other the life forms and her operation base better, because they are conserved in the evolution of the species (Lemos et al.,2003).

Now most of the diseases is provoked by several virus types that possess all their functions found in his genetic material and understanding his DNA better the answer can be foreseen for the cure of these diseases.

With elaboration of algorithms Brute Force, Knuth Morris and Pratt, Iterated Local Search and Boyer Moore heuristic, these algorithms will make the marriage patterns in nucleic acids filaments indicate the amount of find patterns, the pattern's position and the spend time for the search.

This work has as objective the presentation these algorithms used for search of patterns in filaments of DNA with the intention of foreseeing which the best algorithm will find more number of patterns in smaller time.

Key-words: Nucleic Acids, Brute Force, Knuth Morris e Pratt, Iterated Local Search, Boyer Moore, DNA, RNA, Heurístic.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

DNA	Ácido Desoxirribonucléico
RNA	Ácido Ribonucléico
ILS	Iterated Local Search
KMP	Knuth Morris e Pratt
HBM	Heurística de Boyer-Moore

SUMÁRIO

1 INTRODUÇÃO

O ramo da Genética se concentra em estudar tudo sobre os genes, por que são deles que se originam todas as formas de vidas existentes e suas funções indispensáveis para se formar os indivíduos ou provocar doenças em uma determinada espécie, com as características fundamentais das quais são formadas.

A genética é uma ciência que cresce tanto quanto a computação nos dias atuais, e pensando nisso, com este trabalho pretende-se conciliar um estudo genético sobre os filamentos de ácidos nucléicos e transportá-lo para computação como forma de solucionar um desafio presente não só na genética mas em várias áreas relacionadas ao casamento de padrões.

Este trabalho tem como objetivo apresentar os algoritmos mais conhecidos e utilizados na computação para se encontrar casamento de padrões em fitas de ácidos nucléicos de vírus e bactérias, a fim de entender melhor sobre eles, para maiores estudos de soros e vacinas e dentre eles detectar qual é o melhor algoritmo a ser utilizado como forma de otimização de tempo.

Vale ressaltar que o trabalho está ligado a busca de padrões em cadeias genéticas de vírus por que são deles que originam a maioria das doenças, porém a busca por padrões em filamentos com estes algoritmos também pode ser utilizadas em cadeias maiores e até mesmo em outras áreas da computação como, por exemplo, busca de padrões em textos ou dicionários.

Primeiramente será mostrado o contexto biológico dos ácidos nucléicos, as formas, estruturas e como é feito seu mapeamento, depois como irá ser utilizado na base de dados para servir como entrada para os algoritmos elaborados.

A seguir um estudo sobre algoritmos, heurísticas e metas-heurísticas explicando seu funcionamento e alguns de seus principais métodos utilizados e conhecidos na atualidade e como eles servirão para a busca dos padrões na cadeia.

Por fim será apresentada a metodologia utilizada para elaboração dos algoritmos, a forma de entrada da cadeia, no que se consistirá o ponto de parada do algoritmo, e o tempo que eles gastaram para se fazer a busca na cadeia por padrões com o respectivo número de padrões encontrados.

Os resultados serão as análises do melhor ou melhores algoritmos que encontraram o maior número de padrões em menor tempo possível, indicando também posteriores temas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Em todo mundo a tecnologia está em constante evolução, diante disto estão em crescimento acelerado os estudos no ramo da computação e da biologia principalmente na área de genética trabalhando com as cadeias dos genes a fim de descobrirem mais a respeito dos seres vivos.

2.1 ÁCIDOS NUCLÉICOS

Ácidos nucléicos são macromoléculas de enorme importância biológica, todos os seres vivos possuem dois tipos de ácidos nucléicos, chamados de ácido desoxirribonucléico (DNA) e ácido ribonucléico (RNA) (Robertis,2001). Os vírus contêm apenas um destes tipos de ácidos nucléicos, DNA (ácido desoxirribonucléico) ou RNA (ácido ribonucléico).

Os ácidos nucléicos estão relacionados com o controle do funcionamento e da estrutura das células e com os mecanismos da hereditariedade (FAVARETTO, 2005). Em si, os ácidos nucléicos são longas cadeias formadas por unidades chamadas nucleotídeos, ligadas entre si pelo fosfato e por pontes de hidrogênio. A Figura 1 ilustra a forma de um nucleotídeo.

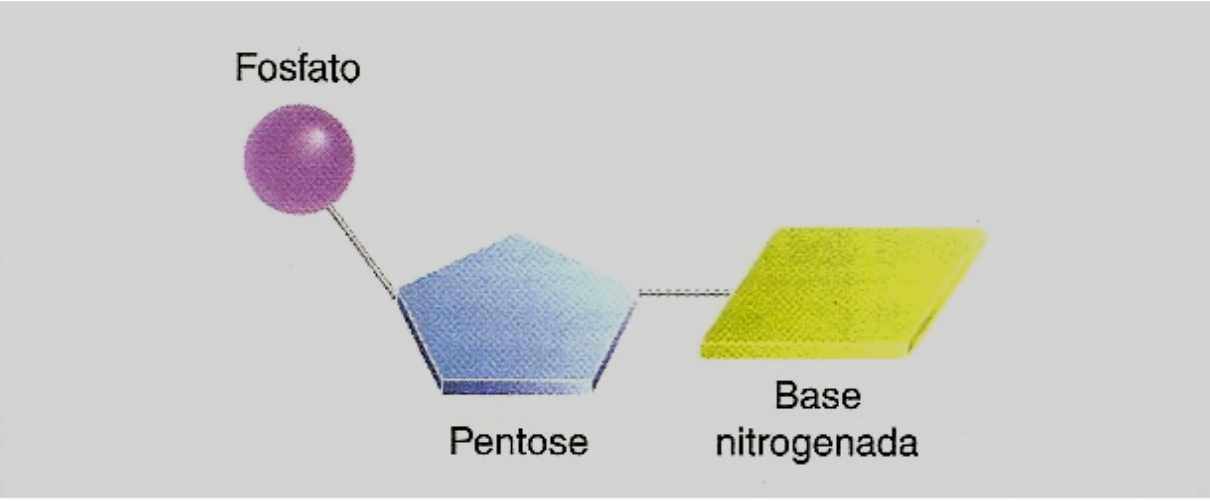

Figura 1 – Nucleotídeo
Fonte: Favaretto (2005), pág. 91

A constituição de um nucleotídeo é de um fosfato, uma pentose e uma base nitrogenada que são ou adenina, ou timina, ou citosina ou guanina para o DNA e adenina, ou uracila, ou citosina ou guanina para o RNA.

O código genético é universal, ou seja, as mesmas partes em diferentes organismos codificam os mesmos aminoácidos (FAVARETTO, 2005).

Aminoácidos são as substâncias codificadas por uma trinca de bases nitrogenadas chamadas códon. Um códon é a permutação das quatro bases nitrogenadas seja do DNA (adenina, timina, citosina, guanina) ou do RNA (adenina, uracila, citosina, guanina) em conjuntos de três (LOPES e ROSSO, 2006).

2.1.1 Ácidos Desoxirribonucléicos

O ácido desoxirribonucléico ou DNA é constituído de quatro tipos de bases nitrogenadas diferentes que são: adenina, timina, citosina, guanina. Este ácido é responsável por comandar a produção de proteínas e controlar a estrutura e o funcionamento das células. Pode-se duplicar-se, gerando cópias perfeitas de si mesmas, no entanto, sua seqüência de nucleotídeos sofre ocasionalmente alterações (mutações) que podem modificar a síntese de proteínas (FAVARETTO, 2005).

Watson e Crick (apud FAVARETTO, 2005) demonstraram que a molécula de DNA é uma dupla hélice formada por duas cadeias de nucleotídeos ligadas entre si por pontes de hidrogênio que conectam as bases nitrogenadas dos nucleotídeos das cadeias, na forma de adenina com timina e citosina com guanina. A cadeia que se sucede é alternadamente, a desoxirribose de um nucleotídeo e o grupo fosfato do seguinte.

A Figura 2 ilustra o esboço do ácido desoxirribonucléico de acordo com o modelo de Watson e Crick (apud FAVARETTO, 2005).

Figura 2 - DNA em forma de dupla hélice
Fonte: http://www.cca.ufscar.br/lamam/disciplinas_arquivos/sequenciamento.htm

2.1.2 Ácidos Ribonucléicos

O Ácido Ribonucléico, ou RNA, é constituído de quatro bases nitrogenadas diferentes que são: adenina, uracila, citosina, guanina. Este ácido é encontrado livre, associado ao DNA ou constituindo os nucléolos. O RNA é uma cadeia única, que pode formar dobras mantidas por pontes de hidrogênio entre suas bases nitrogenadas (FAVARETTO, 2005).

Há três tipos de RNA:

- RNA mensageiro: É geralmente uma longa cadeia formada a partir de uma das cadeias de DNA, que lhe serve de molde.
- RNA transportador: É constituído por uma cadeia dobrada sobre si mesma.
- RNA ribossômico: Junto com proteínas é um componente estrutural dos ribossomos.

O processo de produção do RNA, também chamado processo de transcrição, é catalisado pela enzima RNA polimerase (LOPES e ROSSO, 2006). Este processo acontece por um rompimento do filamento duplo das pontes de hidrogênio do DNA, em seguida as bases de nucleotídeos de RNA emparelham-se com as suas complementares em uma das cadeias de DNA e os nucleotídeos são unidos por ação da enzima RNA polimerase, formando-se o RNA, então o RNA desprende-se restabelecendo as pontes de hidrogênio entre as duas cadeias do DNA.

As principais diferenças entre o DNA e o RNA são apresentadas na Tabela 1. Nesta tabela são descritos, na primeira coluna, cinco fatores que descrevem as

diferenças entre estes ácidos, que são: Pentose, Bases Nitrogenadas, Cadeias, Localização e Funções. Na segunda e terceira colunas são relacionadas as características do DNA e do RNA de acordo com sua descrição, respectivamente de acordo com (LOPES e ROSSO, 2006).

Tabela 1 - Diferenças entre DNA e RNA

Diferenças	DNA	RNA
Pentose	Desoxirribose	Ribose
Bases nitrogenadas	Adenina, Timina, Citosina, Guanina	Adenina, Uracila, Citosina, guanina
Cadeias	Geralmente duas	Geralmente uma
Localização	Principalmente no núcleo	Núcleo e citoplasma
Funções	Controle da estrutura e da atividade celular	Síntese de proteínas

2.1.3 Mapeamento dos Ácidos Nucléicos

"O seqüenciamento de DNA é um processo que determina a ordem dos nucleotídeos (blocos que constituem a molécula de DNA) em uma amostra." (GENESIS, 2003).

Atualmente o mapeamento de filamentos de ácidos nucléicos se dá a partir de enzimas de restrição (enzimas capazes de reagir com os ácidos nucléicos) que cortam as moléculas no trecho onde ela reage. Depois eles são separados e colocados em uma placa de gelatina especial (gel de agarose, onde são colocadas as enzimas de restrição) que passa por uma técnica chamada eletroforese (uma corrente elétrica passa neste gel a fim de se alinhar à cadeia cortada). Após a separação, adicionam sondas (trechos conhecidos de DNA ou ddNTPs) com bases nitrogenadas radiativas que se emparelham com os segmentos isolados de DNA, marcando-os com a radiatividade. Este processo é apresentado na Figura 3.

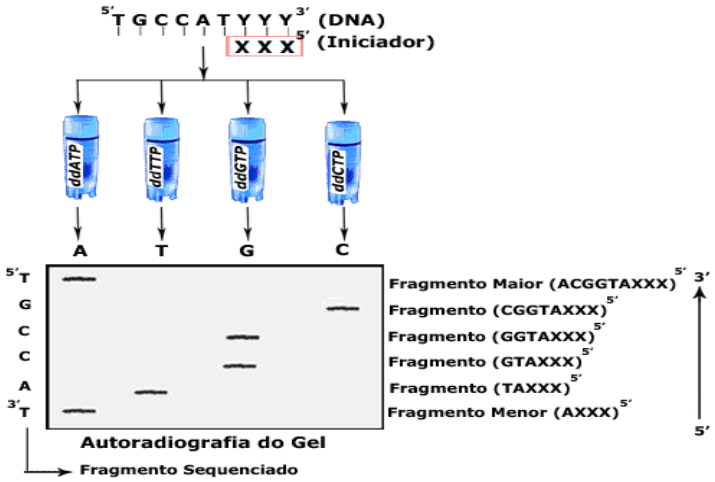

Figura 3 - Folha com parte seqüenciadas
Fonte: GÊNESIS, Laboratório. **Seqüenciamento de DNA.** 2003. [on line].

A seguir a placa de gelatina é colocada, no escuro, sob um filme fotográfico virgem. Após algum tempo, cada série deixa sua impressão no filme. O resultado é semelhante a um código de barras (LOPES et al., 2006).

O método pode ser automatizado através de maquinaria apropriada gerenciada por computadores com programas que lêem sequencialmente e identificam os produtos. Isto permitirá executar o processo em grande escala. Neste caso utilizam-se simultaneamente os quatro didesoxinucleotídeos terminadores (ddNTPs ou enzimas radiativas) marcados por fluorescência, como ilustrado na Figura 4. (GENESIS, 2003).

Ainda na Figura 4, cada reação (A,T,G,C – Adenina, Timina, Guanina e Citosina) utilizou um fluorocromo (cada base nitrogenada radiativa possuía uma diferença de comprimento de onda) diferente, os produtos podem ser reunidos e a eletroforese destes realizada em um único canal do gel de seqüenciamento. O sinal fluorescente diferencial emitido por cada fragmento, após iluminação com um feixe de laser, identificará os produtos baseado na diferença de comprimento de onda. A luz emitida é detectada por escaneamento do gel e a seqüência deduzida por computador. Variáveis mais modernas, consequentemente mais rápidas e poderosas, incluem a robotização total do processo com a inclusão das etapas de purificação e da reação de síntese da cadeia do DNA (GENESIS, 2003).

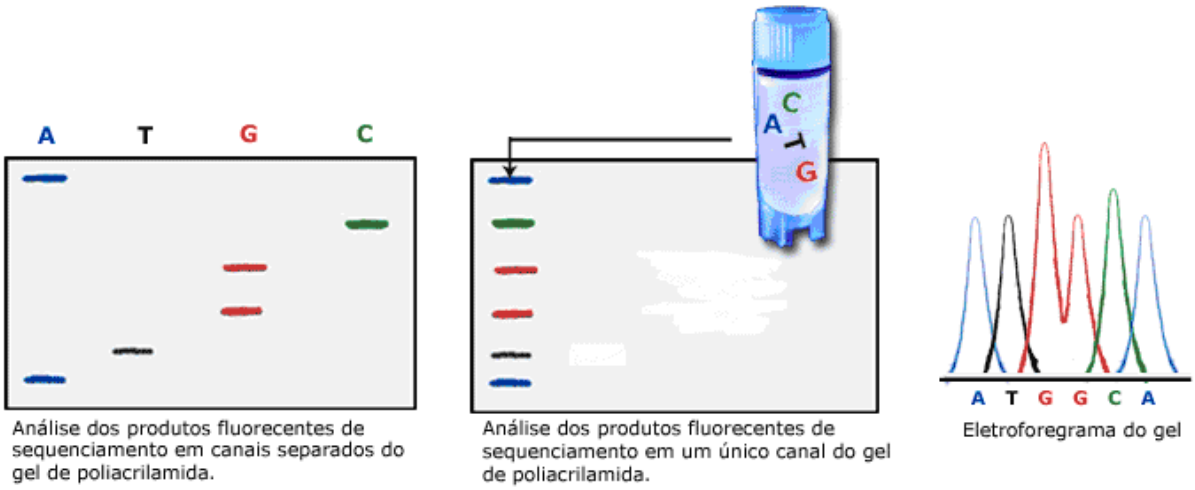

Figura 4 - Análise dos produtos fluorescentes
Fonte: GÊNESIS, Laboratório. **Seqüenciamento de DNA.** 2003. [on line].

2.1.4 Vírus

Os vírus são seres simples, formados basicamente por uma cápsula protéica (capsídeo), envolvendo o material genético(LOPES e ROSSO, 2005). O conjunto capsídeo juntamente com o material genético forma o nucleocapsídeo. Dependendo do tipo de vírus, o seu material genético pode ser o DNA ou RNA.

Os vírus são menores que as células conhecidas e são visíveis apenas ao microscópio eletrônico. Alguns vírus são chamados de envelopados porque possuem um envelope lipoprotéico proveniente da membrana plasmática da célula hospedeira, como ilustrado pela Figura 5.

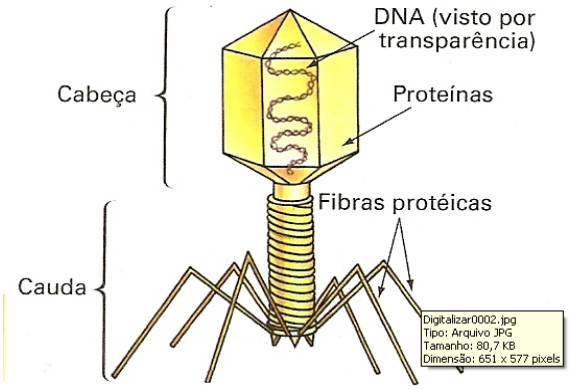

Figura 5 - Esquema de um vírus
Fonte: LOPES e ROSSO. (2005), pág. 191.

Há inúmeras espécies de vírus no planeta e um dos modos de classificação é de acordo com sua espécie de infecção, que são elas (LOPES e ROSSO, 2005):

- Bacteriófagos: São vírus que infectam bactérias.
- Vírus de plantas: São vírus que infectam somente plantas, porém o efeito mais comum das infecções virais nas plantas é o declínio da taxa de crescimento.
- Vírus de animais: São vírus que infectam as células animais causando doenças, câncer e até a morte.

Os vírus possuem um ciclo de vida bastante simples que começa com a infecção de uma célula hospedeira, logo depois ele injeta na célula seu material genético e faz mais cópias de si mesmo dentro da célula, que por sua vez não suporta a quantidades de vírus e sofre, o que na biologia é chamado de, *lise* (rompimento de sua membrana), e o ciclo recomeça.

O material genético dos vírus geralmente não ultrapassa algumas centenas de milhares o que é relativamente baixo em comparação com as outras espécies, então, daí a importância de se estudar seu genoma como forma de se aprender mais sobre eles e tentar conseguir soluções para curas de certas doenças. A Tabela 2 mostra o tamanho dos vírus em comparação com o de outras espécies de acordo com (SANTOS, 2005).

Tabela 2 - Tamanho aproximado dos Ácidos Nucléicos para cada espécie.

Espécie	**Nome popular**	**Tamanho total do genoma**
Bacteriófago	Vírus	$5x10^4$
Escherichia coli	Bactéria	$5x10^6$
Saccharomyces cerevisiae	Levedura	$5x10^7$
Caenorhaditis elegans	Verme	$5x10^8$
Drosophila melanogaster	Mosca	$5x10^8$
Homo sapiens	Homem	$5x10^9$

3 ALGORITMOS

Os processos formais de cálculos exigem o estudo e compreensão do problema em questão, e para resolvê-los muitas vezes é necessário se criar passos para se chegar a determinada solução que se deseja. Em vista disto se entra no conceito de elaboração de algoritmos.

3.1 INTRODUÇÃO

"Os algoritmos podem ser vistos como uma seqüência de ações para a obtenção de uma solução para um determinado tipo de problema.” (ZIVIANI, 2002).

Da mesma forma que pessoas utilizam métodos para resolverem determinado problema os computadores também utilizam, e os algoritmos representam uma forma didática de entendimento dos passos que o computador vai realizar para resolver tal tarefa depois de convertido o algoritmo para uma linguagem de máquina, executável.

Os algoritmos podem ser utilizados para resolverem vários problemas da computação, tais como criação de estruturas, listas lineares, árvores, busca seqüencial e binária, processamento de cadeias, etc.

3.2 COMPLEXIDADE DE ALGORITMOS

A complexidade, em síntese, é descrever o quão eficiente o algoritmo é para solucionar determinado problema. “O objetivo desses métodos é determinar uma expressão matemática que traduza o comportamento de tempo de um algoritmo.” (SZWARFITER, 1994).

Para se obter uma expressão matemática de complexidade de um algoritmo são feitas algumas simplificações de acordo com as regras, como seguem (SZWARCFITER, 1994):

- Algoritmos com quantidades reduzidas de dados não serão considerados. Somente o comportamento assintótico será avaliado, ou seja, a expressão matemática fornecerá valores de tempo que serão válidos unicamente quando a quantidade de dados correspondente crescer o suficiente.
- Não serão consideradas constantes aditivas ou multiplicativas na expressão matemática obtida. Isto é, a expressão matemática obtida será válida, a menos de tais constantes.

As principais funções de complexidade de acordo com ZIVIANI (2002) são descritas abaixo:

- f(n) = $\mathcal{O}(1)$. Algoritmos de complexidade constante: O uso do algoritmo independe do tamanho da entrada. Neste caso as instruções são executadas um número fixo de vezes.
- f(n) = $\mathcal{O}(\log n)$. Algoritmos de complexidade logarítmica: Este tempo de execução ocorre em algoritmos que resolve um problema transformando-o em problemas menores, geralmente partindo-o ao meio e descartando uma metade.
- f(n) = $\mathcal{O}(n)$. Algoritmos de complexidade linear: É a melhor situação de um algoritmo que tem que processar **n** elementos de entrada e produzir **n** elementos de saída.
- f(n) = $\mathcal{O}(n \log n)$. Algoritmos de complexidade logarítmica: Este tempo de execução ocorre em algoritmos que resolvem um problema transformando-o em problemas menores e depois ajuntando as soluções.
- f(n) = $\mathcal{O}(n^2)$. Algoritmo de complexidade quadrática: Ocorre em quando itens são processados aos pares, muitas vezes um anel dentro do outro.
- f(n) = $\mathcal{O}(n^3)$. Algoritmo de complexidade cúbica: São úteis para resolver somente problemas pequenos, podem ocorrer em ciclos de um anel dentro de outro que está em outro.

- f(n) = $\mathcal{O}(2^n)$. Algoritmo de complexidade exponencial: Não são úteis sob o ponto de vista prático. Se **n** aumenta o tempo de execução aumenta exponencialmente.

3.3 PROCESSAMENTO DE CADEIAS

Entende-se por cadeia uma seqüência qualquer de elementos denominados caracteres escolhidos de um conjunto denominado alfabeto. Os caracteres não possuem relações estruturais entre si, a não ser a sua ordem seqüencial. (SZWARCFITER, 1994).

Elas aparecem na computação em várias áreas seja em processamento de textos, envio de mensagens eletrônicas, criptografias de arquivos e senhas, sequenciamento de ácidos nucléicos para biologia computacional, dentre vários outros.

Existem diversos problemas a se considerar em processamento de cadeias, mas dois deles são muito importantes e extensamente utilizados, os quais são compactação de dados e casamentos de cadeias (SZWARCFITER, 1994).

3.3.1 Compactação de Dados

Entende-se por Compactação de Dados, o processo de diminuir o tamanho do arquivo original como forma de economia de espaço. Atualmente os programas utilizados pelas pessoas consomem muita memória o que torna a necessidade de compactar dados indispensáveis. Este recurso é muito utilizado se há necessidade de transmissão de dados via rede mundial de computadores (internet), pois quanto menor o arquivo mais rápido será sua transmissão, dentre outras utilidades.

Existem dois algoritmos básicos de compactação de dados que são:

- **Algoritmo de Freqüência de Caracteres**: É um algoritmo utilizado quando se possui uma grande quantidade de símbolos idênticos. “Determina-se a quantidade de símbolos idênticos existentes no texto. Cada uma das subseqüências máximas de símbolos idênticos do texto é substituída por um número indicando a freqüência do símbolo em questão.” (SZWARCFITER, 1994).
- **Algoritmo de Huffman**: Dado o texto com o conjunto de símbolos e conhecida a freqüência de cada símbolo no texto. “Deseja-se atribuir um código a cada símbolo, de modo a compactar o texto todo” (SZWARCFITER, 1994).

3.3.2 Casamento de Cadeias

O casamento de cadeias é o processo de verificar se uma cadeia qualquer se encontra dentro de uma outra cadeia. “Sejam X e Y duas cadeias, o problema do casamento de cadeias consiste em verificar se Y é sub-cadeia de X.” (SZWARCFITER, 1994).

3.3.3 Algoritmo Força Bruta

Dada duas cadeias X e Y, para verificar se Y é sub-cadeia de X, e em caso positivo localizar Y em X. “A idéia consiste em comparar Y com a sub-cadeia de tamanho *m* de X, que se inicia no caractere onde há os valores *v* possíveis da cadeia total” (SZWARCFITER, 1994). Neste caso ***v*** é a posição inicial de procura, e faz-se a comparação pela posição ***v*** com a primeira posição da sub-cadeia até o limite da sub-cadeia (***m)*** ou até o padrão não coincidir, caso encontre o padrão, retorne sua posição, do contrário, incremente ***v*** e retorne a pesquisa.

“É simples analisar um pior caso deste algoritmo. Ele ocorre quando, em cada cadeia de tamanho *m* de X coincide com Y em todas as posições exceto a última. Ou seja, a complexidade é O (m x n).” (SZWARCFITER, 1994).

Geralmente este algoritmo é utilizado em procura de palavras de dicionários, textos quando os mesmos não são muito grandes onde seu tempo de processamento quase não interfere na busca, porém se o texto tem um tamanho elevado não se aconselha a utilizar o Força Bruta pois pode haver um grande tempo perdido na busca pelo padrão.

3.3.4 Algoritmo Knuth, Morris e Pratt (KMP)

O algoritmo KMP é uma melhora do algoritmo força bruta pelo fato de quando não possuir casamento o algoritmo força bruta volta à posição do vetor principal onde iniciou o casamento ocasionando muitas comparações desnecessárias.

“O KMP evita exatamente o caminho de volta, sendo, portanto mais eficiente.” (CARVALHO et al.,1998).

O funcionamento do algoritmo começa com um pré-processamento da palavra a ser pesquisada no texto, estudam-se as suas repetições e daí faz-se o seguinte passo que é o algoritmo de busca que no caso é o algoritmo força bruta.

Caso o algoritmo KMP faça a pesquisa e não encontre o casamento ele não retorna a posição onde iniciou o casamento, mas retorna a pesquisa no padrão a ser procurado até o limite de repetições acrescentado de um e então continuam as comparações no vetor principal.

Como ele não faz a volta sua complexidade é $\mathcal{O}$(n), mas como se pode notar ele é um bom algoritmo para textos com palavras repetidas que no pré-processamento irão ser identificadas, já em textos sem muitos caracteres repetidos sua complexidade se aproxima a do algoritmo força bruta na ordem de $\mathcal{O}$(m x n).

3.4 HEURÍSTICAS

> Heurística é uma técnica que busca alcançar uma boa solução utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou a otimalidade da solução encontrada ou, ainda, em muitos

> casos, ambas, especialmente nas ocasiões em que essa busca partir de uma solução viável próxima ao ótimo. (GOLDBARG e LUNA, 2000).

Heurísticas são métodos estabelecidos para se solucionar determinado problema quando ele é, por sua vez, bem mais complexo do que outro problema típico. Ao se lidar com problemas que possuem ordem de tempo ou complexidade exponencial elevados, talvez não consiga obter a melhor solução em um tempo hábil, por isso usamos heurísticas que nos dão uma solução que não seja necessariamente a melhor, mas é uma solução boa para um determinado tempo desejado, ou seja, proporciona solução ótima ou aproximada e em tempo computacional viável.

3.4.1 Heurísticas Construtivas

Heurísticas construtivas são métodos responsáveis por construir solução inicial de determinado problema, para que as heurísticas de busca possam iniciar seus processamentos, ou seja, fornece solução inicial para uma heurística de busca poder executar.

O processo de construção de heurísticas envolve a soma de componentes em uma solução – inicialmente vazia – até que esta esteja completa (Fernandes et al., 2010).

As heurísticas construtivas podem ser gulosas ou aleatórias. A heurística gulosa faz sua execução escolhendo sempre o melhor caminho no momento da sua decisão, já a aleatória escolhe sempre aleatoriamente até encontrar uma solução inicial factível.

3.4.2 Heurísticas de Busca Local

A heurística de busca local é responsável por melhorar os resultados encontrados. A busca local como o próprio nome diz faz uma busca em volta da solução encontrada a fim de melhorar ainda mais o resultado obtido e como conseqüência seu campo de busca se restringe somente em volta dos ótimos locais encontrados.

“O objetivo da Busca Local é caminhar, de vizinho a vizinho, a cada iteração, visando melhorar uma solução já obtida.” (Blum e Roli, 2003).

3.5 METAHEURÍSTICAS

Uma metaheurística é um conjunto de conceitos que pode ser utilizados para definir métodos aplicáveis a um extenso conjunto de problemas (SUCUPIRA, 2004).

Ou seja, uma metaheurística é um algoritmo aproximado que combina algoritmos construtivos, estratégias de busca local, estratégias para escapar de ótimos locais, dentre outras características particulares de cada metaheurística (FERNANDES et al., 2009).

Existem várias metaheurísticas, cada uma com uma particularidade, dentre elas podemos citar: *Ant Colony Optimization* (Algoritmo Colônia de Formigas), *Simulated Annealing* (Simulação por Resfriamento), *Tabu Search* (Busca Tabu), Iterated Local Search (Busca Local Iterativa), Algoritmos Genéticos, dentre outros. O objeto de pesquisa deste trabalho abordará somente as metaheurísticas *Iterated Local Search* (Busca Local Iterativa) e heurística de Boyer-Moore.

3.5.1 Busca Baseada em Trajetória

Percorre o espaço de busca levando em conta, fundamentalmente, a “vizinhança” da solução em mãos, definida como o conjunto de soluções que podem ser obtidas a partir da aplicação de algum operador à solução atual (SUCUPIRA, 2004).

3.5.1.1 Iterated Local Search - ILS

O algoritmo *Iterated Local Search*, ou Busca Local Iterativa, tenta realizar uma espécie de busca por entornos nos ótimos locais, ou seja, ele cria perturbações (modificações ou troca de vizinhança) para escapar de ótimos locais, ou visitar outros ótimos locais (FERNANDES et al., 2009). Ou seja, ele é um algoritmo de busca em volta da sua posição atual se ela for um local ótimo, para este algoritmo há uma chance maior de se encontrar a solução ao redor dos ótimos locais do que em outras partes do problema.

De acordo com FERNANDES (2009) o algoritmo ILS tem o seguinte procedimento mostrado na Figura 6.

```
Algoritmo ILS()
1  inicio
2      s_0 ← GeraSolucaoInicial;
3      s ← BuscaLocal(s_0);
4      repita
5          s_1 ← Perturbacao(s, historico);
6          s_2 ← BuscaLocal(s_1);
7          s ← CriterioAceitacao(s_1, s_2, historico);
8      ate que criterio de parada seja satisfeito;
9      retorne s
10 fim.
```

Figura 6 - Algoritmo ILS
Fonte: Fernandes (2009).

O algoritmo ILS gera uma solução inicial e faz a pesquisa em volta desta solução a fim de encontrar o resultado, caso não encontre ele cria uma perturbação (ou gera outra solução) guarda-a no histórico de passagens e refaz a pesquisa em volta desta nova solução e confere se ela é aceitável, o procedimento é repetido até que algum critério de parada seja estabelecido ou satisfeito.

Analisando o algoritmo ILS, ele foi adaptado para se fazer busca por padrões da seguinte forma: Uma solução foi gerada aleatoriamente, verificou-se se era um ótimo local analisando se havia algum caractere em determinada posição do padrão, por fim fez-se a busca em volta do local até que a solução fosse encontrada, ou determinado critério de parada tivesse atingido.

3.5.1.2 Heurística de Boyer-Moore (HBM)

O algoritmo de Boyer-Moore foi elaborado em 1977 por Robert S. Boyer e J. Strother Moore com o objetivo de fazer busca de padrões e sua principal idéia é fazer uma procura da direita para a esquerda no padrão a ser buscado (CARVALHO et al.,1998).

O algoritmo posiciona o padrão sobre o caractere mais à esquerda no texto, e faz uma busca da direita para a esquerda. Se não ocorrer nenhuma diferença, então o padrão foi encontrado, caso contrário ocorrerá uma mudança na posição do padrão que é movido para a direita antes que uma nova comparação seja feita. (CARVALHO et al.,1998).

Para calcular a mudança e os caracteres a serem saltados da pesquisa o algoritmo Boyer-Moore possui duas heurísticas que são a heurística do bom sufixo e a heurística do mau caractere e operam independentes e em paralelo.

Se ocorrer um erro no padrão a ser procurado cada heurística propõe um tamanho para ser incrementado e a heurística de maior valor é atribuída ao avanço.

Quando um erro ocorre, a heurística do mau caractere usa a informação sobre onde o mau caractere do texto ocorre no padrão para propor uma mudança. (CARVALHO et al.,1998).

Quando encontra o padrão diferente do texto na posição do padrão e do texto mais o padrão e o e o tamanho do avanço, onde o tamanho do padrão é menor o valor da posição do padrão, então a heurística do bom sufixo assegura que pode avançar um determinado valor em relação ao texto.

O algoritmo de Boyer-Moore é um algoritmo eficiente, mas de acordo com CARVALHO (1998) necessita de algumas exigências para seu desempenho ser satisfatório que são:

- O texto tem que ser grande.
- O padrão deve ter um tamanho considerável(mais de 10 caracteres).

3.5.2 Busca Populacional

Lidam com uma população de soluções, que evolui, principalmente, através de interação entre seus elementos. Esta iteração acontece de acordo com os padrões de cada metaheurística específica.

3.5.2.1 Algoritmos Genéticos

“Os algoritmos genéticos constituem métodos de busca baseados em mecanismo de seleção e evolução natural.” (GOLDBARG et al. 2000).

Em síntese se pode definir o algoritmo genético com as seguintes características (GOLDBARG et al. 2000):

- Operam em um conjunto de pontos (denominados população) e não a partir de pontos isolados.
- Operam em um espaço de soluções codificadas e não diretamente no espaço de busca.
- Necessita como informação somente o valor de uma função objetivo.
- Usam transições probabilísticas e não regras determinísticas.

4 METODOLOGIA

O objetivo do trabalho é indicar qual o algoritmo consegue mapear o maior número de padrões em menor tempo possível, fazendo a análise do número total de padrões semelhantes em cadeias distintas e analisando o tempo gasto por eles.

Os algoritmos Força Bruta, KMP e HBM conseguem mapear todos os padrões existentes no texto, para o ILS o número de padrões que ele conseguir mapear será inferido o tempo em relação ao número total de padrões semelhantes, ou seja, caso exista 100 padrões semelhantes nas cadeias e ele consiga mapear 50 em 30 segundos, será aplicada a regra de quanto tempo ele gastaria para mapear os 100 padrões que é 60 segundos.

4.1 DESENVOLVIMENTO

O desenvolvimento do projeto constituiu na produção de um algoritmo para gerar cadeias aleatórias de DNA ou RNA e guardar essas cadeias em arquivos para se fazer os testes, e na implementação de quatro algoritmos de busca que foram o algoritmo Força Bruta, Knuth Morris e Pratt, heurística de Boyer-Moore e o Busca Local Iterativa (ILS) para se fazer a pesquisa nas cadeias geradas.

A escolha dos algoritmos se deu pelos seguintes fatos sobre cada um deles:

- O algoritmo Força Bruta foi um dos primeiros a serem utilizados na pesquisa por padrões.
- Para (CARVALHO et al.,1998) o algoritmo KMP poderia ser associado a uma cadeia de DNA .
- As exigências do funcionamento do HBM em princípio se enquadram na pesquisa por padrões em cadeias de ácidos nucléicos.
- O algoritmo ILS é uma heurística de busca que será avaliada para ver o seu desempenho no casamento de padrões em ácidos nucléicos.

Todos os algoritmos foram implementados na linguagem de programação C.

4.2 PADRÕES UTILIZADOS

De acordo com DEUSDADO (2008) tomando como exemplo o genoma humano mais de sua metade é composta de seqüências repetitivas, se algumas dessas seqüências que se repetem possuem uma função determinada então tais elementos são denominados motifs.

O nome dos motifs (apud DEUSDADO,2008) está relacionado com o tamanho das suas seqüências que se repetem. Os motifs com sequências repetidas, que são superiores a casa dos milhares, são denominados satélites. Minisatélites são o conjunto de seqüências menores que a dos satélites e perfazendo a ordem de centenas a milhares e Microsatélites são o conjunto de seqüências mínimas, ou seja, ordem de dezenas e raramente excedendo a centenas de repetições.

Os padrões estarão em uma cadeia e serão procurados em outra cadeia auxiliar para a pesquisa seguindo o tamanho dos motifs na ordem de dezena, centena e milhar, ou seja, microsatélites, minisatélites e satélites.

O tamanho do alfabeto de pesquisa é exatamente a número de bases nitrogenadas existentes nos filamentos de ácidos nucléicos que são quatro, adenina, citosina, timina e guanina para o DNA e adenina, uracila, citosina e guanina para o RNA.

As cadeias utilizadas possuem tamanhos variáveis de 10.000, 100.000, e 1.000.000 de caracteres, porque se referem ao tamanho aproximado de vírus e bactérias existentes no mundo. As cadeias com 10.000 caracteres são denominadas cadeias pequenas, as cadeias com 100.000 caracteres são denominadas cadeias médias e são denominadas cadeias grandes as cadeias com 1.000.000 caracteres.

Na realidade os padrões são partes da segunda cadeia que se deseja procurar, por exemplo, para uma cadeia de tamanho 10.000 um padrão de tamanho 10 seria uma seqüência, nesta cadeia, de 10 caracteres.

4.3 TECNOLOGIA UTILIZADA

Os testes foram efetuados em um computador Intel Celeron de 2.53 GHz, memória RAM de 1.60 Gb, disco rígido de 80 Gb executando no sistema operacional Linux e utilizando o aplicativo KATE para executar os testes na linguagem C.

4.4 PROCEDIMENTOS DE TESTES

O objetivo dos programas é mapear o número de seqüências idênticas entre duas cadeias com tamanho dos padrões de 10, 100 e 1.000 caracteres e tamanho das cadeias com 10.000, 100.000 e 1.000.000 de caracteres, ou seja, para cada conjunto de duas cadeias de mesmo tamanho, mostrar o número de padrões semelhantes encontrados nas cadeias com o respectivo tempo de busca para a pesquisa dos padrões como, por exemplo, para duas cadeias de tamanho 10.000 foram feitos testes com o algoritmo Força Bruta para se encontrar o número de padrões com tamanho 10 e o tempo de processamento para esta busca, o número de padrões com tamanho 100 e o tempo de processamento para esta busca e finalmente, o número de padrões com tamanho 1.000 e o tempo de processamento para esta busca. Este processo também foi realizado com os algoritmos Knuth Morris e Pratt, Boyer-Moore e *Iterated Local Search*.

Na procura por padrões, se um algoritmo encontra o padrão procurado, independente do lugar, ele retorna a posição onde o padrão estava e procura por outro padrão da cadeia, não mais procurando aquela mesma parte no restante da cadeia, caso não encontre o padrão ele retorna um valor inválido e procura pelo próximo padrão.

Os quatro algoritmos receberam notas que serão atribuídas a escala 3, 2, 1 e 0 para os algoritmos que conseguirem o primeiro, segundo, terceiro e quarto lugar respectivamente nas cadeias pequenas, médias e grandes e em tamanho de 10, 100, e 1.000 caracteres.

O resultado será a soma dos pontos conseguidos em cada etapa de teste e os algoritmos serão colocados em ordem decrescente do número de pontos.

5 RESULTADOS

Como solução obtida através dos testes dos algoritmos em cadeias de ácidos nucléicos descritas neste trabalho, chegou-se aos seguintes resultados apresentados nas Tabelas 3, 4, 5, 6, 7 e 8 e nos gráficos ilustrados nas Figuras 7, 8, 9, 10, 11 e 12.

5.1 RESULTADOS OBTIDOS PARA AS CADEIAS PEQUENAS

A seguir é mostrada a tabela do tempo, Tabela 3, de busca e a tabela do número de padrões encontrados, Tabela 4, indicando o desempenho dos algoritmos em cadeias de 10.000 caracteres.

Tabela 3 - Tempo de busca para cadeia pequena

TAMANHO DA CADEIA	TAMANHO DO ALFABETO	TAMANHO DO PADRÃO	ALGORITMOS e TEMPO DE BUSCA (segundos)			
			FORÇA BRUTA	KMP	HBM	ILS
10.000 c	4 c	10 c	1.09 s	1.32 s	0.38 s	0.57 s
		100 c	1.53 s	1.82 s	0.70 s	0.73 s
		1.000 c	1.29 s	0.95 s	0.68 s	0.79 s

Tabela 4 - Padrões encontrados para cadeias pequenas

NÚMERO DE PADRÕES SEMELHANTES EM CADEIAS PEQUENAS		ALGORITMOS COM NÚMEROS DE PADRÕES ENCONTRADOS			
TAMANHO DO PADRÃO	NÚMERO DE PADRÕES SEMELHANTES	FORÇA BRUTA	KMP	HBM	ILS
10 c	7103 p	7.103 p	7.103 p	7.103 p	4.077 p
100 c	0 p	0 p	0 p	0 p	0 p
1.000 c	0 p	0 p	0 p	0 p	0 p

A Tabela 3 mostra que os algoritmos Força Bruta, KMP,HBM e ILS gastaram 1.09, 1.32, 0.38 e 0.57 segundo para mapearem os padrões semelhantes de tamanho 10, 1.53, 1.82, 0.70 e 0.73 segundo para mapearem padrões com tamanho 100 e 1.29, 0.95, 0.68 e 0.79segundo para mapearem padrões com tamanho 1.000, e a Tabela 4 mostra que os algoritmos Força Bruta, KMP e HBM encontraram 7.103 padrões e o algoritmo ILS encontrou 4.077 padrões com tamanho 10, e que nenhum algoritmo encontrou padrões com tamanho 100 ou 1.000 respectivamente.

A performance ilustrativa de cada um deles é ilustrada nas figuras 7 e 8, que seguem:

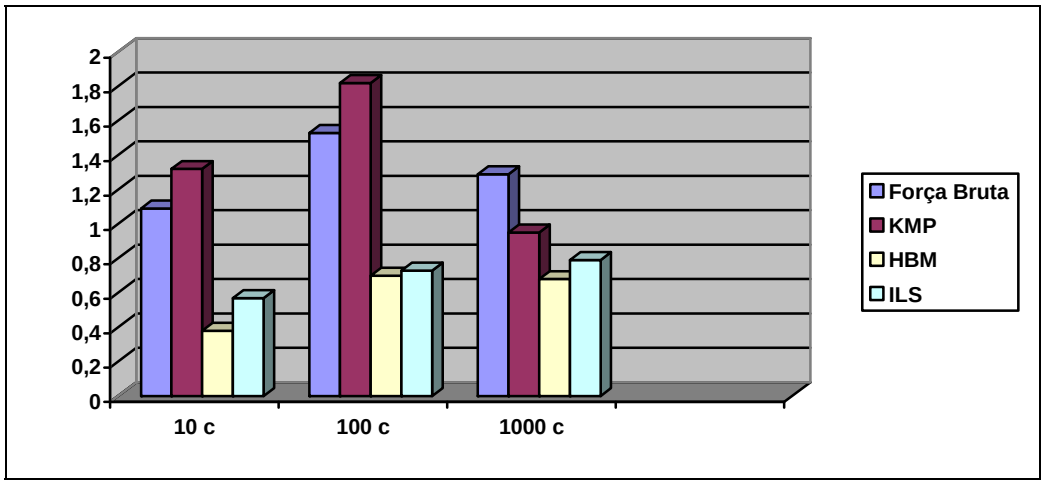

Figura 7 - Gráfico da tabela 3

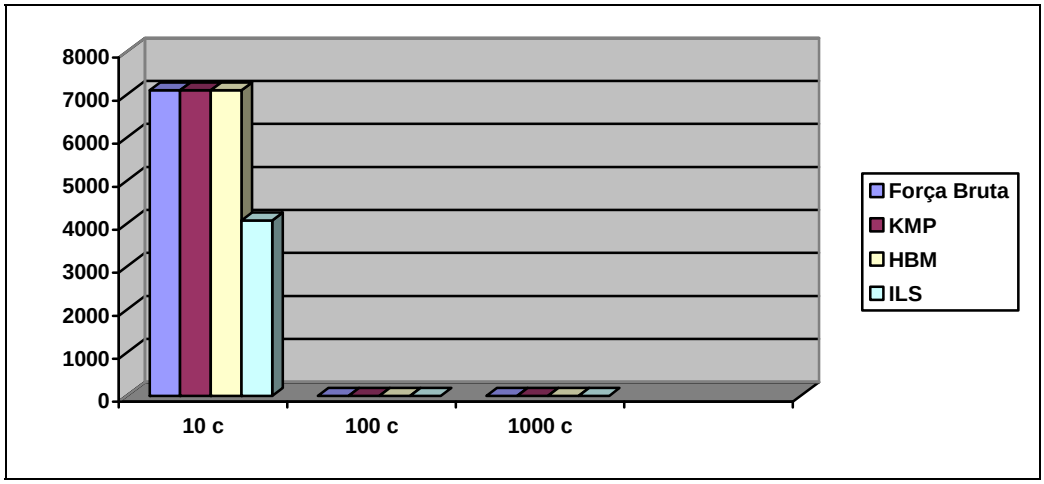

Figura 8 - Gráfico da tabela 4

O ILS conseguiu mapear aproximadamente 57,40% dos padrões com tamanho 10, não conseguiu mapear padrões com tamanho 100 e 1.000, então feito isto somente na primeira ocasião se utilizará a regra de proporção matemática. O que ele demoraria aproximadamente 0.99 segundos para mapear todos os padrões de tamanho 10.

Observando os gráficos e a tabela pode perceber que para cadeias pequenas com padrões de tamanho 10, o HBM ficou em primeiro lugar seguido do ILS, Força Bruta e KMP. Para padrões com tamanho 100 o HBM em primeiro seguido do ILS, Força Bruta e KMP, e para padrões com tamanho 1.000 o HBM em primeiro seguido do ILS, KMP e Força Bruta.

Os resultados parciais dos algoritmos em cadeias pequenas estão ilustrados na Tabela 5.

Tabela 5-Resultados parciais dos algoritmos para cadeias pequenas

ALGORITMO	PONTOS OBTIDOS NA CADEIAS PEQUENAS
FORÇA BRUTA	2
KMP	1
HBM	9
ILS	6

5.2 RESULTADOS OBTIDOS PARA CADEIAS MÉDIAS

A seguir é mostrada a tabela do tempo de busca, Tabela 6, e a tabela do número de padrões encontrados, Tabela 7, indicando o desempenho dos algoritmos em cadeias de 100.000 caracteres.

Tabela 6 - Tempo de busca para cadeia média

TAMANHO DA CADEIA	TAMANHO DO ALFABETO	TAMANHO DO PADRÃO	ALGORITMOS e TEMPO DE BUSCA(segundos)			
			FORÇA BRUTA	KMP	HBM	ILS
100000 c	4 c	10 c	33.04 s	35.22 s	6.54 s	27.14 s
		100 c	137.58 s	161.30 s	56.77 s	65.26 s
		1000 c	141.40 s	163.78 s	57.03 s	66.03 s

Tabela 7 - Padrões encontrados para cadeias médias

NÚMERO DE PADRÕES SEMELHANTES EM CADEIAS MÉDIAS		ALGORITMOS COM NÚMEROS DE PADRÕES ENCONTRADOS			
TAMANHO DO PADRÃO	NÚMERO DE PADRÕES SEMELHANTES	FORÇA BRUTA	KMP	HBM	ILS
10 c	99.289 p	99.289 p	99.289 p	99.289 p	71.619 p
100 c	22.458 p	22.458	22.458 p	22.458 p	17.482 p
1.000 c	21.558 p	21.558	21.558 p	21.558 p	16.799 p

A Tabela 6 mostra que o algoritmo Força Bruta gastou 33.04, 137.58 e 141.40 segundos, o KMP 35.22, 161.30 e 163.78 segundos, o HBM 6.54, 56.77 e 57.03 segundos e o ILS gastou 27.14, 65.26 e 66.03 segundos para mapearem os padrões semelhantes de tamanho 10, 100 e 1.000, respectivamente.

A Tabela 7 mostra que os algoritmos Força Bruta, KMP e HBM encontraram 99.289, 22.458 e 21.558 padrões e o algoritmo ILS encontrou 71.619, 17.482 e 16.799 padrões semelhantes com tamanho 10, 100 e 1000, respectivamente.

A performance ilustrativa de cada um deles é apresentada nas Figuras 9 e 10:

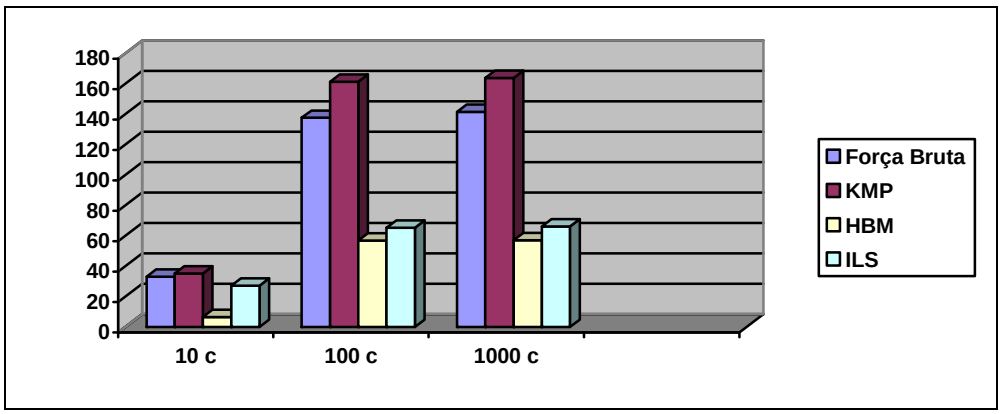

Figura 9 - Gráfico da tabela 5

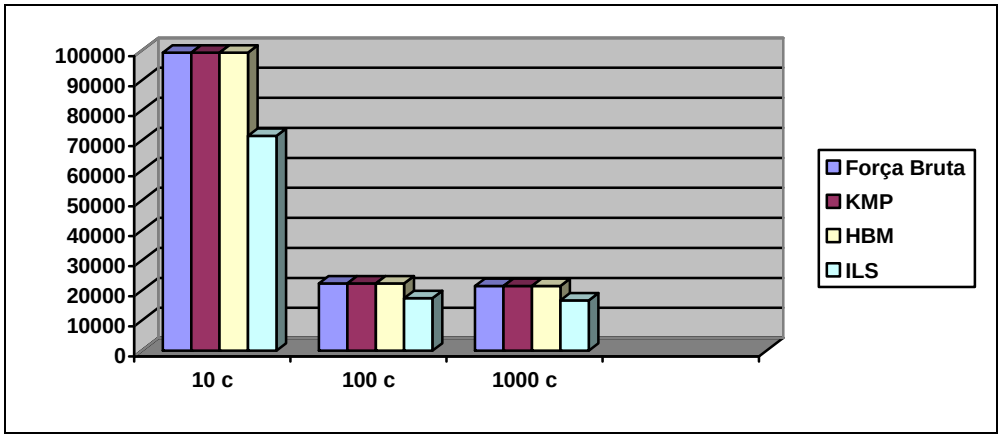

Figura 10 - Gráfico da tabela 6

O ILS conseguiu mapear aproximadamente 72,13% dos padrões com tamanho 10, 77.84% dos padrões com tamanho 100 e 77.92% dos padrões com tamanho 1.000, então feito isto aplicando a regra de proporção matemática nos três resultados se obteria 37.62, 83.83 e 84.73 segundos para mapear todos os resultados de tamanho 10,100 e 1.000 respectivamente.

Observando os gráficos e a tabela pode perceber que para cadeias pequenas com padrões de tamanho 10, o HBM ficou em primeiro lugar seguido do Força Bruta, KMP e ILS. Para padrões com tamanho 100 o HBM em primeiro seguido do ILS, Força Bruta e KMP, e para padrões com tamanho 1.000 o HBM em primeiro seguido do ILS, Força Bruta e KMP.

Os resultados parciais dos algoritmos em cadeias médias estão ilustrados na Tabela 8:

Tabela 8-Resultados parciais algoritmos para cadeias médias

ALGORITMO	PONTOS OBTIDOS NA CADEIAS MÉDIAS
FORÇA BRUTA	4
KMP	1
HBM	9
ILS	4

.

5.3 RESULTADOS OBTIDOS PARA CADEIAS GRANDES

A seguir é mostrada a tabela do tempo de busca, Tabela 9, e a tabela do número de padrões encontrados, Tabela 10, indicando o desempenho dos algoritmos em cadeias de 1000000 caracteres.

Tabela 9 - Tempo de busca para cadeia grande

TAMANHO DA CADEIA	TAMANHO DO ALFABETO	TAMANHO DO PADRÃO	ALGORITMOS e TEMPO DE BUSCA(segundos)			
			FORÇA BRUTA	KMP	HBM	ILS
1000000 c	4 c	10 c	1683.53s	1914.64s	532.43s	1840.26s
		100 c	3337.65s	3467.95s	1132.65s	3557.18s
		1000 c	3391.73s	3625.41s	1212.87s	3776.93s

Tabela 10 - Padrões encontrados para cadeias grandes

TAMANHO DA CADEIA	TAMANHO DO ALFABETO	TAMANHO DO PADRÃO	ALGORITMOS E NÚMEROS DE PADRÕES ENCONTRADOS			
			FORÇA BRUTA	KMP	HBM	ILS
1000000 c	4 c	10 c	999.990p	999.990p	999.990p	750.261p
		100 c	999.900p	999.900p	999.900p	743.614p
		1000 c	999.000p	999.000p	999.000p	750.135p

A tabela 9 mostra que o algoritmo Força Bruta gastou 1683.53, 3337.65 e 3391.73 segundos, o KMP 1914.64, 3467.95 e 3625.41 segundos, o HBM 532.43, 1132.65 e 1212.87 segundos e o ILS gastou 1840.26, 3557.18 e 3776.93 segundos para mapearem os padrões semelhantes de tamanho 10, 100 e 1.000, respectivamente.

A Tabela 10 mostra que os algoritmos Força Bruta, KMP e HBM encontraram 999.990, 999.900 e 999.000 padrões e o algoritmo ILS encontrou 750.261, 743.614 e 750.135 padrões semelhantes com tamanho 10, 100 e 1000, respectivamente.

A performance ilustrativa de cada um deles é apresentada nas Figuras 11 e 12, respectivamente, que seguem:

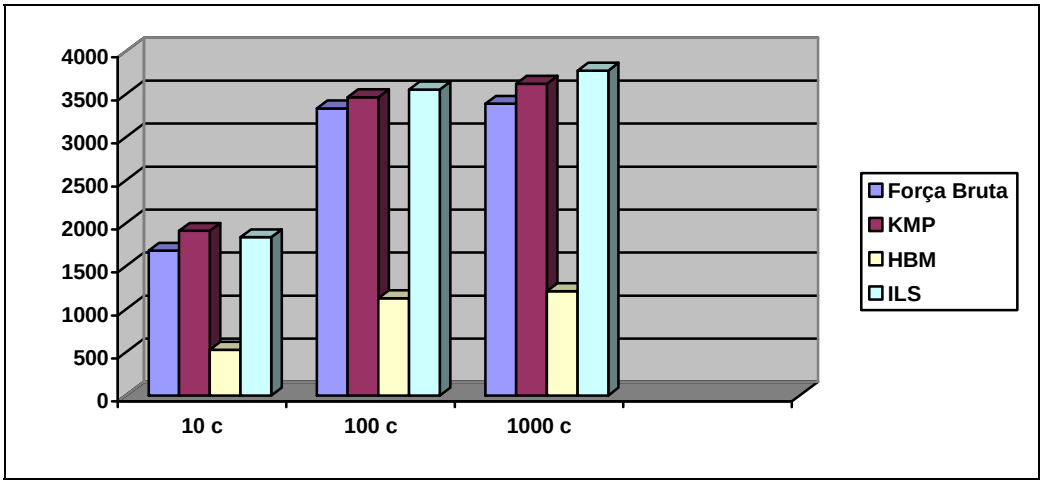

Figura 11 - Gráfico da tabela 7

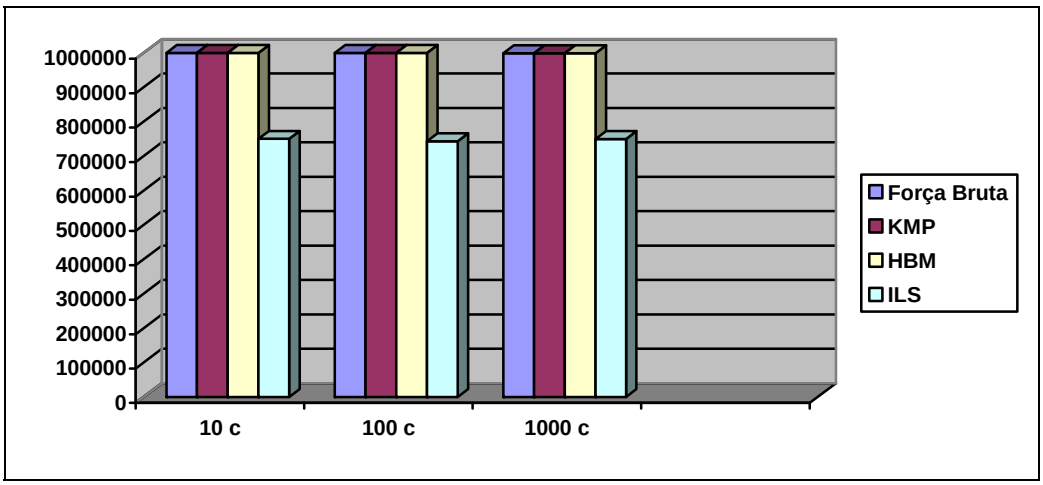

Figura 12 - Gráfico da tabela 8

O ILS conseguiu mapear aproximadamente 75,02% dos padrões com tamanho 10, 74.37% dos padrões com tamanho 100 e 75.08% dos padrões com tamanho 1.000, então feito isto aplicando a regra de proporção matemática nos três resultados se obteria 2452.80, 4783.15 e 5029.96 segundos para mapear todos os resultados de tamanho 10,100 e 1.000 respectivamente.

Observando os gráficos e a tabela pode perceber que para cadeias grandes os algoritmos mantiveram as mesmas ordens para padrões de tamanho 10,100 e 1.000 o HBM ficou em primeiro lugar seguido do Força Bruta, KMP e ILS.

Os resultados parciais dos algoritmos em cadeias médias estão ilustrados na Tabela 11:

Tabela 11-Resultados parciais dos algoritmos para cadeias grandes

ALGORITMO	PONTOS OBTIDOS NA CADEIAS GRANDES
FORÇA BRUTA	6
KMP	3
HBM	9
ILS	0

5.4 ANÁLISE DOS RESULTADOS DOS PROGRAMAS

O resultado total é a soma dos pontos obtidos em todos os tipos de cadeias nos três tipos de padrões como ilustrado na Tabela 12.

Tabela 12-Resultado total dos algoritmos nas três cadeias

ALGORITMO	PONTOS OBTIDOS EM TODAS AS CADEIAS
FORÇA BRUTA	12
KMP	5
HBM	27
ILS	10

De acordo com os resultado da tabela acima se pode notar que o melhor algoritmo foi o HBM seguido do Força Bruta, ILS e KMP

Como se pode observar nos gráficos anteriores os algoritmos Força Bruta, KMP e HBM conseguiram encontrar número de padrões iguais, mesmo com tamanhos diferentes em tempos variados e o algoritmo ILS encontrando pouco mais de 70 por cento dos padrões.

O algoritmo HBM conseguiu mapeá-los com menos de 40 por cento do tempo que os algoritmos Força Bruta e KMP gastariam para mapearem cadeias acima de 100000.

O algoritmo ILS não obteve um melhor desempenho pelo fato de suas passagens serem guardadas em um histórico o qual ele necessita verificar-lo para

observar se aquele trecho do código onde ele indicou já foi pesquisado consumindo mais tempo para fazer isto e segundo é que sua busca local é feita por casamento igual ao método força bruta.

O algoritmo KMP não obteve o desempenho esperado mesmo o alfabeto das cadeias de ácidos nucléicos sendo pequenas.

6 CONCLUSÃO

Observando os resultados dos algoritmos descritos anteriormente, constata-se que o melhor algoritmo para casamento de padrões em filamentos de ácidos nucléicos é a heurística de Boyer-Moore, principalmente pelo fato de sua eficiência que de acordo com (CARVALHO et al.,1998) é dada pelo padrão ter um tamanho considerável acima de 10 caracteres o que é o ideal e está relacionado com o tamanho de motifs de pesquisa e o seu tempo de busca por padrões foi o menor dos algoritmos apresentados em todos os testes feitos.

(CARVALHO et al.,1998) diz que o algoritmo KMP mostra-se mais eficientes com caracteres repetidos podendo ser associado a uma cadeia de DNA, todavia com os resultados obtidos verificou-se que seu desempenho é inferior ao algoritmo Força Bruta o que demonstra que apesar do alfabeto de uma cadeia de DNA ou RNA ser pequena (apenas 4 caracteres) ele talvez necessite de um alfabeto ainda menor como por exemplo um alfabeto ternário ou binário ou as cadeias a serem avaliadas necessitem ser ainda maior na ordem dezenas ou centenas de milhões.

O algoritmo Força Bruta obteve a segunda posição sendo mais eficaz que o KMP e o ILS, mesmo sendo considerado um algoritmo com ordem de complexidade alta.

O ILS se mostrou mais eficiente que o KMP para cadeias pequenas e médias, mas caiu seu desempenho à medida que a cadeia cresceu isso se deve ao fato do ILS pesquisar no histórico a passagem por determinado local toda vez que faz uma pesquisa o que consome tempo considerável. Contudo obteve a terceira posição na soma total dos pontos nos três tipos de cadeias.

Diante do exposto, verifica-se que de acordo com o aumento do tamanho das cadeias ou o número de cadeias a serem pesquisadas, percebe-se que o tempo de busca por padrões aumenta exponencialmente, daí exige-se a necessidade de utilizar heurísticas para se obter resultados aceitáveis em um tempo considerável.

7 TRABALHOS FUTUROS

De acordo com o trabalho verifica-se a exigência de uma melhoria em algoritmos ou a distribuição do processamento destes programas, então fica com sugestão de trabalhos futuros a composição de algoritmos que realizem casamento de padrões em sistemas paralelos ou *threads* para tentativa de um melhor desempenho e utilização de um número menor de máquinas a fim de otimizar o tempo em cima das heurísticas, pois foram as que obtiveram as melhores performances.

Por fim, realizar um estudo algorítmico na tentativa de melhorar a performance dos métodos implementados, bem como a implementação de outras estratégias heurísticas, tais como Algoritmos Genéticos e Algoritmos de Colônia de Formigas.

REFERÊNCIAS BIBLIOGRÁFICAS

BLUM, C.; ROLI, A. **Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Computing Surveys.** 2003.

CARVALHO, Paulo Sérgio R.; OLIVEIRA, Deive D.; GUARACY, Alessandra; Gomes, Alisson; CONCEIÇÃO, Fernando César da; FREIRE, Joseane; OLIVEIRA, Jones Alburque. **Um Estudo Teórico e Experimental em Algoritmos Clássicos de Busca de Texto**. UFLA-MG Lavras, 1998.

DEUSDADO, Sérgio Alípio Domingues. **Análise e comparação de seqüências genómicas.** Dissertação de Doutorado. Universidade do Minho São Paulo, 2008.

Favaretto, José Arnaldo; Mercadante, Clarinda. **Biologia**. 1ªed. São Paulo: Editora Moderna, 2005.

FERNANDES, Filipe C.; SOUZA, Sérgio R. de; BORGES, Henrique Elias; SILVA, Maria Amélia L.;Gama, Pedro H. A. **Uma Adaptação da metaheurística Iterated Local Search para a resolução do problema de roteamento de veículos com janela de tempo**. Anais do XVIII Congresso Brasileiro de Automática – CBA, Bonito-MS, 2010.

FERNANDES, Filipe C.; SOUZA, Sérgio R. de; SILVA, Maria Amélia L.; BORGES, Henrique E. **Arquitetura Multiagentes Baseada em Metaheurísticas para Solução de Problemas de Otimização Combinatória**. Anais do IX Simpósio Brasileiro de Automação Inteligente – SBAI, Brasília-DF, 2009.

GÊNESIS, Laboratório. **Seqüenciamento de DNA.** 2003. [on line]. Disponível em: http://www.cca.ufscar.br/lamam/disciplinas_arquivos/sequenciamento.htm. Último acesso em: 16, junho, 2010.

GOLDBARG, Marco César; LUNA, Henrique Pacca L. **Otimização Combinatória e Programação Linear.** 5ª Tiragem Rio de Janeiro: Editora Campus, 2000.

ROBERTIS, E. M. F. de; HIB, Jose. **Bases da Biologia Celular e Molecular.** 3ª ed. Rio de Janeiro: Editora Guanabara Koogan, 2001.

LEMOS, Melissa; ARAGÃO, Marcus V. S. Poggi de; CASANOVA, Marco Antônio. **Padrões em Biossequências**. PUC-Rio Inf.MCC17/03 Junho, 2003.

LOPES, Sônia; ROSSO, Sérgio. **Biologia**. 1ªed. São Paulo: Editora Saraiva, 2005. 2ª tiragem, 2006.

SANTOS, Jean Rodrigo dos; PINTO, Rafael Caetano; SANTOS, Valdemárcio Fernandes dos. **Comparação de Seqüências de DNA em sistemas distribuídos clusterizados.** Centro Universitário FIEO São Paulo, 2005.

SUCUPIRA, Igor Ribeiro. **Métodos Heurísticos Genéricos**. USP São Paulo**,** 2004.

SZWARCFITER, Jayme Luiz; MARKENZON, Lílian. **Estrutura de Dados e Seus Algoritmos.** 2ª ed. Rio de Janeiro: Editora LTC - Livros Técnicos e Científicos Editora S.A., 1994.

ZIVIANI, Nivio. **Projeto de Algoritmos: Com Implementações em Pascal e C.** 6ª reimpressão da 1ª ed. São Paulo: Editora Pioneira, 2002.