Regiões Ortólogas em Múltiplos Genomas

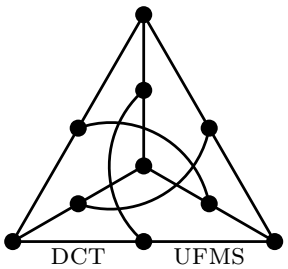
Luciana Montera

Dissertação de Mestrado (04-2004)

Orientação: Prof. Dr. Nalvo Franco de Almeida Junior

Área de Concentração: Biologia Computacional

Dissertação apresentada ao Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul como parte dos requisitos para a obtenção do título de mestre em Ciência da Computação.



Departamento de Computação e Estatística
Centro de Ciências Exatas e Tecnologia
Universidade Federal de Mato Grosso do Sul
28 de abril de 2004

Regiões Ortólogas em Múltiplos Genomas

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Luciana Montera
e aprovada pela comissão julgadora.

Campo Grande, 29 de Abril.

Banca examinadora:

- Prof. Dr. Nalvo Franco de Almeida Junior - orientador (DCT-UFMS)
- Prof. Dr. Maria Emilia Machado Telles Walter (CIC-UnB)
- Prof. Dr. Marcelo Augusto Santos Turine (DCT-UFMS)
- Prof. Dr. Marcelo Henriques de Carvalho (suplente) (DCT-UFMS)

Aos meus pais

Agradecimentos

Como é bom chegar ao final de um trabalho e ter a quem agradecer, pois é um sinal de que temos amigos, e eu, graças a Deus tenho muitos.

Começo agradecendo ao meu orientador, professor Nalvo Franco de Almeida Junior, pela atenção e paciência dedicados a mim, desde a graduação até agora, na conclusão desse trabalho de mestrado. Aos professores Henrique Mongelli, Katia Mara França da Silva, e todos os demais professores e funcionários do DCT, onde sempre fui muito bem acolhida.

Um gradecimento especial aos professores Marcelo Henriques Carvalho e Edna Ayako Hoshino, pela importante colaboração com esse trabalho.

À minha amada família, que mesmo a distância sempre foi e será o meu ponto de apoio.

Não poderia deixar de falar dos grandes amigos que fiz no decorrer do Mestrado. Mais uma vez, a amiga e professora Edna, que com o seu enorme entusiasmo pelos estudos, sempre nos "puxava" para cima nas horas mais difíceis de nossos estudos em grupo. A Claudia e a Sibelis por estarem sempre dispostas a ouvir minhas reclamações e brincadeiras. Aos amigos Erik e Luciano Gonda, pelos convites para tomar tereré em sua sala ou um suco no corredor central, convites estes que na maioria das vezes chegavam em ótima hora.

Aos meus amigos de sala Carlos Juliano e Graziela, ótimas companhias para um dia de estudos. Ao Carlos por ser sempre tão atencioso e disposto a ajudar, além de ser quem, na maioria das vezes, ficava encarregado de buscar água e preparar o tereré nosso de cada dia. Um agradedimento especial a minha grande amiga Graziela, com a qual compartilhei a maior parte dos momentos vividos durante o mestrado e por ser ela quem mais estava presente nos momentos tristes e alegres vividos neste período, dentro e fora da Universidade.

Minhas grandes amigas Silvia e Meyre não poderiam ficar de fora desta lista,

pois elas também fizeram parte desta etapa.

Por fim, agradeço de forma especial ao meu namorado e amigo durante os anos de mestrado e agora meu esposo, Andrés Batista Cheung, pela compreensão, carinho e amor sempre dedicados a mim.

Resumo

Este trabalho apresenta uma metodologia para encontrar regiões contíguas de genes entre vários genomas, que evolutivamente conservam a ordem e o conteúdo gênico. Tais regiões, por terem se preservado ao longo da evolução, são descritas como potenciais detentoras de funcionalidades em organismos procariotos evolutivamente próximos. A principal contribuição do trabalho consiste num algoritmo baseado na abordagem do problema de cliques maximais em grafos para a determinação de regiões de genes conservados entre múltiplos genomas. Um sistema via web foi desenvolvido para permitir ao usuário realizar as comparações entre múltiplos genomas e visualizar graficamente os resultados obtidos.

Abstract

This work presents a methodology for finding contiguous gene regions of several genomes with conserved gene order and content. Because of their conservation through evolution, such regions are putative functional in closely related procaryote organisms. The main contribution of this work is an algorithm based on a maximal clique problem approach to find the regions, from regions found in pairwise genome comparison, besides a web system where the user can choose some organisms to be compared.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Atualmente, os vários projetos de seqüenciamento, dentre eles o Projeto Genoma Humano, tratam de um volume considerável de dados, que precisam ser processados adequadamente de forma a se obter informações biologicamente relevantes. Esse processo inclui várias fases, sendo que a primeira delas corresponde à determinação das seqüências de DNA de um organismo com o objetivo de reconhecer as diversas regiões que compõem essas moléculas. Dentre essas regiões existem aquelas de fundamental importância para a síntese de proteínas pelas células e, conseqüentemente, para o correto funcionamento do organismo. Essas regiões são conhecidas como genes ou regiões codificantes.

O número de genomas completamente seqüenciados está aumentando rapidamente, em particular para o caso de procariotos, tais como bactérias. Para se ter uma idéia, em Março/2004 era 180 o número de genomas de procariotos publicados, e outros 1084 projetos em andamento [21].

Mesmo com os evidentes avanços na área de Bioinformática, a comunidade científica ainda não é capaz de processar eficientemente toda essa massa de informação gerada, tanto quanto é capaz de produzí-la. Uma das fases deste processamento consiste nas análises das seqüências genômicas, visando obter caracterizações funcionais mais detalhadas. Essas análises tomam como base o resultado da anotação dos genes, que consiste em determinar suas respectivas funções, e também a comparação com os outros genomas, no nível de seus genes, mais especificamente no nível de suas proteínas preditas.

A comparação de genomas pode ser vista em dois níveis. Num primeiro nível compara-se diretamente as seqüências de DNA genômico, na tentativa de obter informações relevantes, que possam refletir algum tipo de relacionamento entre as espécies. Essas informações podem vir de similaridades

locais, encontradas por meio de alinhamentos entre trechos de seqüências; da identificação de trechos com alto nível de identidade; de repetições mais significativas encontradas nos genomas, entre outras [1].

Num outro nível, toma-se como informação as localizações exatas dos genes de cada genoma, que geralmente são disponíveis após o seqüenciamento e a anotação de um genoma. A partir dos genes localizados em cada um dos genomas, tenta-se obter um grafo que reflita os relacionamentos entre os diversos genes, tais como blocos de genes similares e rearranjos. Para tanto, necessita-se de um algoritmo eficiente para a construção de tal grafo, levando-se em conta o tamanho de cada gene (aproximadamente 1000 bases) e principalmente o elevado número de genes em um genoma.

Inúmeros trabalhos [1, 33, 35] realizam comparações entre dois genomas e identificam, entre outras coisas, genes homólogos entre eles.

Porém, mais importante que comparar dois genomas é a comparação de vários genomas simultaneamente. Uma técnica atualmente empregada na comparação simultânea entre várias seqüências é a construção de um *alinhamento múltiplo*. Um alinhamento múltiplo deve ser capaz de identificar similaridades menos aparentes em múltiplas seqüências, em detrimento a fortes similaridades aparentes apenas em comparações dois-a-dois. Esse problema, mesmo para seqüências pequenas, é considerado complexo. Assim, busca-se na literatura ferramentas que levam em conta não a seqüência propriamente dita do genoma, mas sim características mais pontuais, mais discretas, que possam ser ao mesmo tempo localizadas em outros genomas e que reflitam de alguma forma o relacionamento entre eles. Isso é possível a partir de comparações realizadas entre os genes dos genomas e a partir de regiões dos dois genomas que envolvem genes onde a ordem e a funcionalidade são preservadas. A determinação de tais regiões, chamadas de *regiões ortólogas*, é um dos objetivos do trabalho publicado por Almeida [1]. As definições de alinhamento múltiplo e região ortóloga serão dadas mais adiante no texto.

A dificuldade em comparar simultaneamente vários genomas reside principalmente na identificação de regiões que acontecem em múltiplos genomas, que passa obviamente pela determinação dos grupos de genes dos vários genomas que preservam a ordem e a vizinhança.

A justificativa do trabalho, portanto, está na necessidade de um ambiente computacional que envolva ao mesmo tempo o aspecto de alinhamento múltiplo de genomas, e que forneça pistas de funcionalidades comuns entre eles, a partir dessas regiões conservadas. Esse ambiente terá como base técnicas de alinhamento múltiplo e comparações dois-a-dois dos genomas.

A principal contribuição deste trabalho consiste num algoritmo, baseado em cliques maximais em grafos, para encontrar regiões que se mantêm conservadas em múltiplos genomas, em termos de conteúdo gênico, a partir de regiões determinadas nas comparações dois-a-dois dos genomas. Além disso, o trabalho propõe um método de visualização de tais regiões múltiplas baseado na construação de alinhamentos múltiplos, assim como um sistema web onde o usuário pode escolher os organismos a serem comparados e informar valores para determinados parâmetros utilizados na comparação. O trabalho resultou na publicação [2].

O restante do texto se encontra organizado da seguinte forma. No capítulo 2 encontram-se as definições básicas sobre Biologia Molecular e Genômica Comparativa, necessárias ao entendimento do problema, além de uma breve introdução aos problemas de alinhamento entre duas ou mais seqüências. No capítulo 3 é descrito o problema de comparação entre dois proteomas e é apresentada a ferramenta EGG [1], utilizada para realizar esta comparação; e também o problema da comparação entre múltiplos proteomas, juntamente com o algoritmo proposto para solucioná-lo. O capítulo 4 apresenta a ferramenta web desenvolvida e detalhes referentes ao banco de dados desenvolvido e às ferramentas utilizadas durante o trabalho. Finalmente, no capítulo 5 apresentamos comentários conclusivos e perspectivas para trabalhos futuros.

Capítulo 2

Preliminares

Neste capítulo apresentaremos uma breve introdução aos conceitos básicos de Biologia Molecular, assim como algumas definições usadas na comparação de seqüências, em especial na comparação de proteomas.

2.1 Conceitos de Biologia Molecular

A célula é a unidade estrutural e funcional básica dos seres vivos, podendo eles ser formados por uma ou várias (milhares) delas. Existem dois tipos de células, as **procarióticas** e as **eucarióticas**. A principal diferença entre elas é a ausência de um envoltório nuclear nas células procarióticas. Dessa forma, os seres vivos podem ser divididos em dois grandes grupos, os **procariotos** (ex.: bactérias e algas azuis) e os **eucariotos** (demais organismos).

Os **ácidos nucléicos** são moléculas de suma importância biológica presentes nas células. Os ácidos nucléicos estão presentes em todos os organismos vivos de duas formas distintas: ácido ribonucléico (**RNA**) ou ácido desoxiribonucléico (**DNA**), sendo eles formados por unidades menores denominadas **nucleotídeos**.

Os nucleotídeos que compõem a molécula de DNA podem ser de quatro tipos diferentes. Eles são formados por açúcar, a *desoxirribose*, um grupo de fosfato e uma molécula chamada **base nitrogenada**. O que difere os quatro tipos de nucleotídeos entre si é a base nitrogenada que os compõem. As bases nitrogenadas presentes na molécula de DNA são: Adenina (**A**), Timina (**T**), Citosina (**C**) e Guanina (**G**). No RNA, o açúcar presente é a *ribose* e a base nitrogenada Tinima é substituída pela base Uracila (**U**).

Para efeitos deste trabalho uma **seqüência** de **DNA** é uma seqüência de letras escrita sobre o alfabeto Σ formada pelas letras `A`, `C`, `G`, `T`. Letras adicionais são admitidas para representar combinações de nucleotídeos, quando não se tem certeza de qual letra deve ocupar uma posição na cadeia. Esse alfabeto é descrito em [23].

O DNA é o principal armazenador de informação genética. A seqüência linear das quatro bases A, T, C e G presentes no DNA de um ser vivo é a responsável pela codificação das proteínas necessárias à sua sobrevivência. O DNA é formado por uma **fita dupla**, denominadas fita + e fita **-**, as quais tem **orientações** opostas e são tais que um `A` sempre pareia com um `T` e um `C` sempre pareia com um `G`. As orientações são ditas serem da extremidade $5'$ para a extremidade $3'$. A figura 2.1 mostra um exemplo de como as fitas se pareiam. Uma fita é dita ser o **complemento-reverso** da outra. O complemento-reverso de um trecho de DNA g é denotado por g^{CR}. Medimos o tamanho de um trecho de DNA de fita dupla pelo seu número de **pares de base**, denotados por **bp** (de *base-pairs*).

```
5'      ...  A A T C G G C A T T C G G T C  ...      3'
             | | | | | | | | | | | | | | |
3'      ...  T T A G C C G T A A G C C A G  ...      5'
```

Figura 2.1: Exemplo da representação da fita dupla do DNA.

Apenas alguns trechos do DNA codificam *proteínas*, e estes trechos são chamados **genes**. As **proteínas** são moléculas constituídas por moléculas de **aminoácidos** e realizam as mais variadas funções no nosso organismo, desde o transporte de nutrientes e metabólitos à catálise de reações biológicas. O arranjo linear, ou seqüência, de resíduos de aminoácidos que constituem uma proteína é chamado de **estrutura primária**, e é esta estrutura que determina a forma e a função da proteína. Os aminoácidos presentes na seqüência se "enrolam" através de ligações peptídicas até um quarto nível de estruturação. Desta forma, descobrir a seqüência de aminoácidos que deu origem a uma proteína é muito importante, já que ela vai determinar sua função.

Para que um gene dê origem a uma proteína é necessário que ocorra uma *tradução* das moléculas de nucleotídeos (presentes nos genes) em moléculas de aminoácidos (presentes nas proteínas). Esta tradução ocorre seguindo o que se chama **Código Genético**, o qual estabelece uma relação entre três nucleotídeos e um aminoácido. Dessa forma temos um conjunto de $3^4 = 64$ triplas possíveis para a codificação de aminoácidos, as quais são denominadas

códons. Apesar de existirem 64 códons diferentes, o número de aminoácidos existentes é de apenas 20, devido ao fato de alguns códons diferentes codificarem um mesmo aminoácido.

2.2 Conceitos utilizados em Genômica Comparativa

Nesta seção introduzimos algumas definições básicas usadas quando se comparam dois proteomas, ou seja, quando se comparam dois genomas através de suas proteínas preditas.

Ao conjunto de genes de um organismo dá-se o nome de **genoma**, e ao conjunto de proteínas preditas pelos genes de um organismo chamamos **proteoma**.

As **coordenadas de um gene** são as posições de seu início e fim na fita +. Assim, um gene da fita − tem como coordenada de início a posição correspondente, na fita +, à sua posição final; e tem como coordenada final, a posição correspondente, na fita +, à sua posição inicial. Apenas uma das fitas é utilizada para representar todos genes de um genoma, a fita +. A ordem em que os genes aparecem na representação é dada pela ordem não-decrescente de coordenadas iniciais. Assim, um proteoma pode ser visto como a seqüência dos genes, mais precisamente as proteínas preditas de um genoma, onde a ordem e a fita em que os genes aparecem no genoma é levada em consideração. A figura 2.2 mostra de forma bem simples a representação gráfica de um proteoma. A partir deste ponto usaremos indistintamente os termos gene e proteína predita.

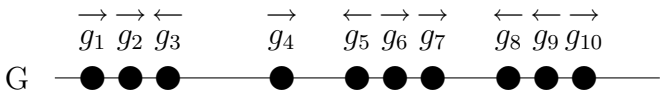


Figura 2.2: Proteoma de um Genoma G. Cada g_i é um gene.

Os conceitos de homologia, ortologia e paralogia são de extrema importância quando estamos interessados em comparar organismos. Para as definições

a seguir considere os genes g e g' pertencentes ao genoma G e o gene h pertencente ao genoma H:

- g e h são **homólogos** se descendem de um mesmo ancestral comum. Neste caso dizemos que g e h são **ortólogos**;
- g e g' são **homólogos** se descendem de um mesmo ancestral comum. Neste caso dizemos que g e g' são **parálogos**;
- o gene g é específico em relação a H se não existe em H nenhum gene ortólogo a g.

Uma **RGC** (Região de Genes Consecutivos) é um conjunto de genes consecutivos num genoma, de acordo com suas coordenadas de início, independentes da fita. Note que o próprio genoma consiste em uma RGC.

Deste ponto em diante, seguem algumas definições computacionais relevantes para o entendimento da solução proposta para o problema da comparação entre múltiplos proteomas.

Um **grafo** $\mathcal{G}$ é um par formado por um conjunto V de elementos chamados **vértices** e um conjunto E de elementos chamados de **arestas**; é denotado por $\mathcal{G} = (V, E)$. Cada aresta é um par não-orientado de vértices. Assim, $(u, v) = (v, u)$. Um grafo é **simples**, quando (u, v) é tal que $u \neq v, \forall u, v \in V$ e tal que não há arestas múltiplas ligando o mesmo par de vértices. Dizemos que dois vértices u e v são **vizinhos** em $\mathcal{G}$, se $(u, v) \in E$.

Um grafo $\mathcal{G}$ é **bipartido** se V pode ser particionado em dois conjuntos X e Y tais que qualquer aresta (u, v) é tal que: ou $u \in X$ e $v \in Y$, ou $u \in Y$ e $v \in X$.

Um conjunto C de vértices de um grafo $\mathcal{G}$ é uma **clique** se a seguinte propriedade for verdadeira: para todo par (u, v) de vértices distintos em C, existe aresta $(u, v) \in E$. Uma clique C é **maximal** se não existe clique C' que a contenha. As figuras 2.3 e 2.4 apresentam exemplos de uma clique e uma clique maximal, respectivamente, presentes em um grafo G.

2.3 Alinhamento de Seqüências

Nesta seção introduzimos brevemente o conceito de *alinhamento de seqüências*, em especial o conceito de alinhamento múltiplo de seqüências. No contexto

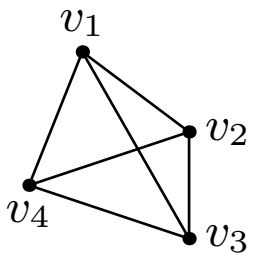


Figura 2.3: Clique formada pelos vértices v_1, v_2 e v_4.

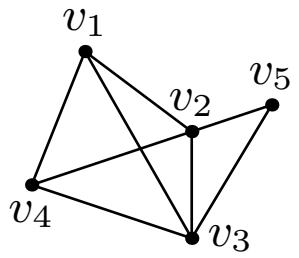


Figura 2.4: Clique maximal formada pelos vértices v_1, v_2, v_3 e v_4.

deste trabalho, o alinhamento múltiplo não é usado para comparar diretamente as seqüências de DNA dos genomas nem dos múltiplos proteomas, mas sim como uma ferramenta de visualização das regiões conservadas entre os vários proteomas.

2.3.1 Alinhamento de duas seqüências

O **tamanho** de uma seqüência (ou cadeia) s, denotado por $|s|$, é o número de símbolos que ela contém. O símbolo que ocupa a posição i é denotado por s_i. Assim, uma seqüência s contém os símbolos $s_1, \ldots, s_{|s|}$. Se $|s| = 0$, dizemos que s é **vazia**.

A comparação entre duas seqüências é um processo que tem como objetivo, entre outros, evidenciar as similaridades e as diferenças existentes entre dois organismos, obtendo assim informações funcionais e evolutivas importantes. Para que essas similaridades e/ou diferenças fiquem aparentes, recorre-se muitas vezes à construção de um *alinhamento* entre as seqüências dos organismos. Desta forma, a similaridade entre duas seqüências é uma medida utilizada para estimar o quão elas são parecidas (próximas). A definição 2.1 apresenta formalmente o problema de alinhamento entre duas seqüências, também chamado de alinhamento *pairwise*.

Definição 2.1 *Dadas as seqüências* $s = s_1 \dots s_m$ *e* $t = t_1 \dots t_n$, *com símbolos pertencentes ao alfabeto* Σ, *e com* $m, n \geq 0$, *um* **alinhamento** *de* s *e* t *é um mapeamento de* s *e* t *nas seqüências* s' *e* t', *respectivamente, cujos símbolos pertencem ao alfabeto* $\Sigma' = \Sigma \cup \{-\}$, *onde o símbolo '*$-$*' é chamado de* **espaço**, *tal que:*

1. $|s'| = |t'| = l$;
2. *a remoção dos espaços de* s' *e* t' *leva a* s *e* t, *respectivamente; e*
3. *não é permitida a condição* $s'_i = - = t'_i$, $1 \leq i \leq l$.

O **valor** *do alinhamento é dado por*

$$\sum_{i=1}^{l} \sigma(s'_i, t'_i). \tag{2.1}$$

onde $\sigma : \Sigma' \times \Sigma' \rightarrow \mathbf{R}$ *é uma função simétrica tal que* $\sigma(a, b)$ *denota o valor do emparelhamento entre o símbolo* a *com o símbolo* b, $\forall a, b \in \Sigma'$. *Apesar de não ser utilizado, o par* $(-, -)$ *deve ser tal que* $\sigma(-, -) = 0$. *A uma seqüência contígua de espaços em* s' *ou em* t' *denominamos* **buraco**. *A cada coluna do tipo* (s'_i, t'_i), *com* $s'_i \neq t'_i$, $s'_i, t'_j \neq -$, *damos o nome de* **substituição**. *Se a coluna for do tipo* (s'_i, t'_i), *com* $s'_i = t'_i$, $s'_i \neq -$, *damos o nome de* **casamento**. *Uma coluna do tipo* $(s'_i, -)$ *recebe o nome de* **remoção** *e uma coluna do tipo* $(-, t'_i)$ *recebe o nome de* **inserção**.

Uma observação pertinente diz respeito à escolha da função σ. Podemos estar interessados em alinhar duas seqüências no sentido de medir a similaridade entre elas, ou de medir a *distância de edição* entre elas. Como já foi dito, na similaridade, estamos interessados em saber o quanto as duas seqüências são parecidas, enquanto que na **distância de edição** procuramos saber o número mínimo de operações (inserções, remoções e substituições de bases) necessárias para transformar uma seqüência na outra. Ou seja, a distância serve para sabermos quão distantes na escala evolutiva estão as seqüências. Muitas vezes estas medidas são intercambiáveis, no sentido de que uma pequena distância significa uma grande similaridade e vice-versa. No entanto, a similaridade é mais flexível: em alguns casos, exige-se que a medida de distância obedeça à desigualdade triangular. No nosso caso estaremos interessados em medir a similaridade entre seqüências. Uma descrição detalhada sobre estas duas abordagens pode ser vista no livro do Setubal e Meidanis [29].

Como dissemos, o alinhamento entre duas seqüências pode evidenciar similaridades existentes entre elas. Dadas duas seqüências, existem diversos alinhamentos possíveis entre elas, porém, na maioria das vezes, estamos interessados em encontrar o alinhamento que melhor representa as similaridades, caso elas existam, ou seja, estamos procurando o **alinhamento ótimo** entre elas. A seguir, apresentamos a definição de alinhamento ótimo entre duas seqüências.

Definição 2.2 *Um* **alinhamento ótimo** *de duas seqüências s e t é definido como um alinhamento que maximiza a soma 2.1, entre todos os possíveis alinhamentos entre s e t.*

Gusfield [15] descreve um algoritmo exato baseado em programação dinâmica que em tempo e espaço $O(mn)$, onde m e n são os tamanhos das seqüências sendo comparadas, encontra o melhor alinhamento entre duas seqüências. Como geralmente as seqüências de DNA são cadeias muito longas, a utilização deste algoritmo torna-se muitas vezes inviável. Faremos a seguir uma breve descrição deste algoritmo.

Sejam duas seqüências s e t de tamanhos m e n respectivamente. Para a construção do alinhamento ótimo entre elas, é utilizada uma matriz A, de tamanho $m \times n$. Seja $A_{i,j}$ o valor de um alinhamento ótimo de $s_1 \dots s_i$ e $t_1 \dots t_j$. Assim, o valor de um alinhamento ótimo de s e t é dado por $A_{m,n}$. A idéia chave da programação dinâmica é resolver o problema mais geral computando todos os valores $A_{i,j}, 0 \leq i \leq m, 0 \leq j \leq n$, em ordem não-decrescente de i e j.

A base do algoritmo é o cálculo de

$$A_{0,0} = 0,$$

$$A_{i,0} = \sum_{k=1}^{i} \sigma(s_k, -) \quad \text{e}$$

$$A_{0,j} = \sum_{k=1}^{j} \sigma(-, t_k).$$

A fórmula geral de recorrência, para $1 \leq i \leq m$ e $1 \leq j \leq n$, é

$$A_{i,j} = \max \begin{cases} A_{i-1,j} + \sigma(s_i, -), \\ A_{i-1,j-1} + \sigma(s_i, t_j), \\ A_{i,j-1} + \sigma(-, t_j). \end{cases}$$

Ao final, temos em $A_{m,n}$ o valor de um alinhamento ótimo entre as seqüências s e t.

Considere a seguinte função σ para o cálculo do valor do alinhamento entre s_i e t_j de duas seqüências s e t:

$$\sigma(s_i, t_j) = \begin{cases} +1, & \text{se } s_i = t_j; \\ -1, & \text{se } s_i \neq t_j; \\ -2, & \text{se } s_i \text{ ou } t_j = \text{“-”}; \end{cases} \tag{2.2}$$

Sejam as seqüências $s = \{ATCCGAT\}$ e $t = \{ACGAAGT\}$. As figuras 2.5, 2.6 e 2.7 mostram três possíveis alinhamentos entre elas. Utilizando a função de valor vista em 2.2, conhecida como função $\mathcal{SP}$ (*Sum of Pair*), temos que o melhor alinhamento entre s e t é aquele mostrado na figura 2.7.

$$\begin{array}{c} A\ T\ C\ C\ G\ A\ T\ - \\ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ | \\ A\ -\ C\ G\ A\ A\ G\ T \end{array}$$

Figura 2.5: Alinhamento entre s e t com valor -4.

$$\begin{array}{c} A\ T\ C\ C\ G\ A\ -\ -\ T \\ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ | \\ A\ -\ -\ C\ G\ A\ A\ G\ T \end{array}$$

Figura 2.6: Alinhamento entre s e t com valor -3.

$$\begin{array}{c} A\ T\ C\ C\ G\ A\ -\ T \\ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ | \\ A\ -\ C\ G\ A\ A\ G\ T \end{array}$$

Figura 2.7: Alinhamento entre s e t com valor -2.

2.3.2 Alinhamento de múltiplas seqüências

A definição de alinhamento múltiplo entre seqüências é uma generalização natural do alinhamento entre duas seqüências, visto na definição 2.1 e é apresentada a seguir, na definição 2.3.

Definição 2.3 *Sejam $s_1, s_2, ..., s_k$ um conjunto de k seqüências escritas sobre um mesmo alfabeto Σ. Um* **alinhamento múltiplo** $\mathcal{M}$ *envolvendo $s_1, s_2, ..., s_k$ é obtido através da inserção de espaços ("-") nas seqüências de tal forma que todas elas fiquem do mesmo tamanho.*

Para um dado conjunto de seqüências existem diversos alinhamentos possíveis entre elas. Como exemplo, sejam as seqüências de aminoácidos $s_1 = \{M, Q, P, I, L, L, L\}$, $s_2 = \{M, L, R, L, L\}$, $s_3 = \{M, K, I, L, L, L\}$ e $s_4 = \{M, P, P, V, L, I, L\}$. A figura 2.8 mostra um possível alinhamento múltiplo $\mathcal{M}$ envolvendo estas quatro seqüências.

```
M Q P I L L L
M L R - L L -
M K - I L L L
M P P V L I L
```

Figura 2.8: Um alinhamento múltiplo entre s_1, s_2, s_3 e s_4.

Assim como no problema de alinhar duas seqüências, temos por objetivo encontrar o alinhamento múltiplo que melhor represente as similaridades existentes entre os organismos envolvidos. Desta forma, uma medida de pontuação deve ser escolhida para medir a qualidade do alinhamento múltiplo encontrado, ou seja, para que se possa atribuir um valor a este alinhamento.

Embora a definição de alinhamento múltiplo seja facilmente estendida a partir da definição de alinhamento entre duas seqüências, o mesmo não acontece com a função de pontuação para um alinhamento múltiplo. Atualmente não há uma função de pontuação bem aceita para alinhamento múltiplo como há para medir a similaridade e a distância de edição entre duas seqüências. Contudo, a função $\mathcal{SP}$ ($Sum\ of\ Pairs$) tem sido muito utilizada para o cálculo do valor a ser atribuido a um alinhamento, devido a sua fácil aplicação ao alinhamento múltiplo.

Definição 2.4 *O valor $\mathcal{SP}$ de um alinhamento múltiplo $\mathcal{M}$ é a soma dos valores dos alinhamentos dois-a-dois induzidos por $\mathcal{M}$. Por alinhamento induzido entende-se todos os possíveis pares de alinhamentos dois-a-dois obtidos a partir de $\mathcal{M}$.*

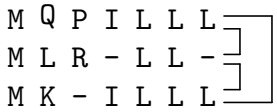


Figura 2.9: Alinhamentos induzidos por $\mathcal{M}$.

A figura 2.9 mostra um alinhamento múltiplo e seus pares de alinhamentos induzidos.

Quanto aos algoritmos existentes para a construção de um alinhamento múltiplo, citaremos dois: um algoritmo exato e outro aproximado.

O algoritmo exato para a solução do problema, ou seja, aquele que encontra o alinhamento de maior valor, consiste em uma extensão do algoritmo exato de programação dinâmica para o alinhamento entre duas seqüências, porém este algoritmo se aplica apenas para seqüências pequenas, devido ao tempo e espaço gastos em sua computação pois, para um conjunto de k seqüências a serem alinhadas, onde a maior delas possui tamanho n, o custo do algoritmo seria $O(n^k)$.

Devido a este fato, algoritmos aproximados são utilizados para solucionar o problema. Descreveremos um destes algoritmos aproximados, conhecido como o método *Alinhamento Estrela*, devido a Gusfield [14].

Sejam $s_1, s_2, ...s_k$ um conjunto de k seqüências que desejamos alinhar. Para construir o alinhamento múltiplo entre elas deve-se primeiramente escolher uma seqüência do conjunto para ser a seqüência centro do alinhamento, chamada s_c. Para se obter a seqüência centro, é necessário calcular o alinhamento ótimo para todos os pares de seqüências do conjunto, e escolher aquela seqüência que maximiza a seguinte somatória:

$$\sum_{i \neq c} similaridade(s_c, s_i). \tag{2.3}$$

Uma vez obtida a seqüência centro, o alinhamento múltiplo é construído progressivamente. O processo se inicia com o alinhamento *pairwise* ótimo entre s_c e uma outra seqüência qualquer, digamos s_1 e s_c, e tem continuidade através da inserção de novas seqüências neste alinhamento considerando apenas o alinhamento ótimo entre a seqüências centro s_c e a nova seqüência a ser inserida. Durante esse processo, buracos podem ser inseridos tanto na seqüência centro como nas outras seqüências já inseridas no alinhamento, para que sua consistência seja mantida.

O custo para obtenção a seqüência centro é de $O(k^2n^2)$ sendo n o tamanho da maior seqüência, enquanto que o custo para a inserção de todas é seqüências no alinhamento é de $O(k^2l)$, onde l é o número máximo de colunas do alinhamento. Assim, o custo total do algoritmo é de $O(k^2n^2 + k^2l)$.

Sejam as seqüências $S_1 = MQPILL$, $S_2 = MLRLL$, $S_3 = MKILLL$ e $S_4 = PMPPVLIL$. Considere S_1 como sendo a seqüência centro para a construção do alinhamento múltiplo entre S_1, S_2, S_3 e S_4. A figura 2.10 mostra os passos executados pelo algoritmo para a construção do alinhamento.

```
M Q P I L L        M Q P I L L L        P M Q P I L L L
M L R - L L        M L R - L L -        - M L R - L L -
                   M K - I L L L        - M K - I L L L
                                        P M P P V L I L
```

Figura 2.10: Alinhamento progressivo entre as seqüências S_1, S_2, S_3 e S_4.

São muitos os textos introdutórios disponíveis para uma melhor compreensão dos conceitos aqui descritos. Setubal e Meidanis [29] e Gusfield [15], por exemplo, apresentam excelente material sobre os mais diversos problemas de Biologia Molecular Computacional, incluindo o tópico de comparação de seqüências. Em particular, os principais conceitos e convenções utilizados neste trabalho estão de acordo com os três primeiros capítulos do livro de Setubal e Meidanis [29] e com Almeida [1].

Para uma leitura sobre tópicos mais avançados de Biologia Molecular Computacional, sugerimos o livro de Pevzner [26] e o livro editado por Salzberg *et al* [27], além das notas escritas por Phil Green [13]. Os livros de Cummings e Klug [6], Lehninger [19] e Lewin [20] também são indicado para conceitos de Biologia Molecular.

Capítulo 3

Comparação de Proteomas

Neste capítulo apresentamos a principal contribuição de nosso trabalho, que consiste num algoritmo para comparar simultaneamente vários proteomas. Em linhas gerais, desenvolvemos um algoritmo que encontra regiões que preservam o conteúdo gênico em vários proteomas a partir das regiões preservadas entre eles, tomadas dois-a-dois.

Antes de mostrarmos nosso algoritmo, vamos descrever como essas regiões entre dois proteomas são encontradas.

3.1 Comparação de dois proteomas

Nossa metodologia para encontrar regiões que se preservam entre vários proteomas tem como base as regiões que se preservam entre os pares de proteomas comparados. Para tanto, fazemos uso de uma ferramenta que compara dois proteomas e que tem como um dos objetivos encontrar estas regiões preservadas.

Esta ferramenta é denominada EGG, de *Extended Genome-Genome Comparison*, descrita por Almeida [1].

Dados dois proteomas G e H, os principais objetivos do EGG são:

- determinar os genes específicos de G em relação a H, e vice-versa;
- determinar os pares de genes ortólogos entre G e H;
- encontrar as regiões de genes próximos que de alguma forma preservam o conteúdo gênico nos dois proteomas;

- encontrar um alinhamento global dos dois proteomas, chamado de espinha dorsal de G e H.

Dos objetivos citados acima, estamos interessados no terceiro. As regiões encontradas por EGG são chamadas de *Regiões Ortólogas* e são definidas adiante no texto. EGG tem sido utilizado com sucesso em vários projetos genoma, como uma ferramenta de análise [7, 11, 30, 31, 39].

É importante notarmos aqui que, apesar de EGG encontrar um alinhamento global entre G e H (o quarto objetivo da lista acima), não estamos interessados em construir alinhamento múltiplo dos proteomas. A razão para isto é, além da ineficiência dos algoritmos para alinhamento múltiplo, o fato de que a espinha dorsal de dois proteomas é bem determinada apenas para dois organismos muito próximos evolutivamente. Certamente um alinhamento múltiplo de proteomas não seria capaz de mostrar regiões menores, com alto nível de conservação.

Antes de descrevermos o funcionamento do EGG, vejamos algumas definições necessárias. O primeiro conceito a ser apresentado é o de *run*, uma vez que este é necessário para a definição de região ortóloga.

Definição 3.1 *Sejam dois genomas G e H. Para uma RGC α de G formada pelos genes $g_i, ..., g_k$ e uma RGC β de H formada pelos genes $h_j, ..., h_l$, tais que $k - i + 1 = l - j + 1$, $k > i$, e $l > j$, dizemos que α e β formam um* **run** *se uma das seqüências de pares de genes ortólogos acontece:*

- $(g_i, h_j), (g_{i+1}, h_{j+1}), ..., (g_k, h_l)$*; ou*
- $(g_i, h_l), (g_{i+1}, h_{l-1}), ..., (g_k, h_j)$.

Quando a primeira situação acontece temos um run **paralelo**, quando a segunda situação acontece temos um run **anti-paralelo**. As desigualdades $k > i$, e $l > j$ nos dizem que um run tem que ter pelo menos dois pares de genes ortólogos.

Agora que já temos o conceito de run, podemos definir o conceito de *região ortóloga* (RO).

Definição 3.2 *Uma* **Região Ortóloga** *R é :*

- *um run com pelo menos M pares de ortólogos; ou*

- *a união de runs de qualquer tamanho, totalizando pelo menos M pares de ortólogos, e cuja distância entre os genes extremos dos runs mais próximos entre si não seja maior que um certo valor fixo k, em número de genes.*

Desta forma, podemos dizer que a estratégia utilizada para a determinação de ROs está baseada na junção de runs próximos. Alguns cuidados devem ser tomados para que as ROs encontradas tenham de fato algum sentido biológico, como por exemplo, a escolha dos valores adequados para M e k.

Testes realizados por Almeida [1] sugerem o valor $M = 3$, para genomas de procariotos, para garantir que um run não seja encontrado ao acaso. Como na junção de runs podem existir buracos tanto no genoma G quanto no genoma H, o valor fixo k é representado por dois valores inteiros: max_large_gap e max_small_gap, os quais representam respectivamente, o valor máximo do maior e do menor intervalo de genes existente entre os runs. Os valores padrões para esses parâmetros são:

$$max\_large\_gap = 5 \text{ e } max\_small\_gap = 2$$

A justificativa para que se permita a existência destes "buracos" em uma região ortóloga, causados por genes não casados nesta região, está no fato de que biologicamente eles podem refletir a remoção, substituição ou a inserção de genes durante o processo de evolução.

A figura 3.1 mostra um exemplo abstrato de uma RO encontrada entre dois proteomas, digamos G e H. Um gene em uma RO pode participar de mais de um par de ortólogos. Este fato, caracterizado pela existência de genes parálogos, será explicado adiante, quando será especificada a forma como os pares de genes ortólgos são determinados.

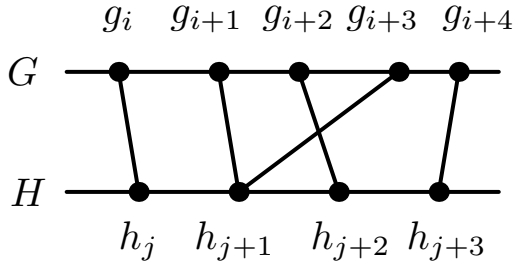


Figura 3.1: Exemplo de uma RO abstrata.

Determinação dos pares de ortólogos

Como vimos, o algoritmo para encontrar o alinhamento ótimo entre duas seqüências é computacionalmente muito caro e por isso, na prática são usados algoritmos aproximados para este fim. Uma das ferramentas mais utilizadas atualmente para realizar buscas em bases de dados e comparação entre seqüências é denominada BLAST - **B***asic* **L***ocal* **A***lignment* **S***earch* **T***ool* [4], e se encontra disponível em *http://www.ncbi.nih.gov/blast/*. Para realizar a comparação entre as seqüências dos genes que compõem os proteomas, a ferramenta *EGG* faz uso do BLAST.

Em linhas gerais, BLAST toma como entrada uma seqüência denominada **query** e retorna como resposta uma lista de *hits*. Um **hit** é uma seqüência encontrada na base de dados que é similar à seqüência query. Juntamente com cada hit são também retornados:

- o alinhamento entre a seqüência query e o hit;
- o **valor** do alinhamento;
- valor de significância estatística, chamado **e-value**, proporcional à probabilidade daquele hit ter sido encontrado ao acaso no banco de dados.

A lista de hits encontrados para uma dada seqüência query encontra-se ordenada em ordem não-decrescente do valor do e-value de cada alinhamento.

Etapas da ferramenta EGG

Passamos agora a descrição da ferramenta EGG, a qual é composta por três grandes fases:

- Fase 1: Comparação dos genes do genoma G contra todos os genes do genoma H, e vice-versa;
- Fase 2: Construção de um grafo bipartido, considerando os resultados obtidos na Fase 1. Neste grafo, os vértices são genes e as arestas representam a relação de ortologia entre genes;
- Fase 3: Determinação de estruturas organizacionais presentes no grafo.

Como resultado da Fase 1 temos para cada gene g do genoma G e h do genoma H uma lista de hits, cada um acompanhado do alinhamento, do valor e do e-value correspondente. É importante lembrar que podemos ter e-values diferentes quando g é encontrado como hit de h e para quando h é encontrado como hit de g. Essa situação pode acontecer quando tratamos de proteomas de tamanhos muito diferentes e com muitos genes parálogos.

Na Fase 2, fase onde as arestas do grafo são determinadas, EGG define o conceito de *match*. Dois genes g e h formam um **match** se g encontrou h como hit com o valor do e-value de no máximo 10^{-5} e vice-versa. Além disso, pelo menos 60% das seqüências de g e h são cobertas pelo alinhamento. Desta forma um match representa uma aresta do grafo. Temos ainda que dois genes g e h formam um **BBH** (**B**idirectional **B***est* **H***it*), quando g encontrou h como o melhor hit (menor e-value) e vice-versa.

Note que para um determinado gene g pertencente ao genoma G, podem ser encontrados mais do que um match no genoma H. Isso explica o fato de existirem duas arestas incidentes no vértice h_{j+1}, como foi visto na figura 3.1.

Os matches entre os genes dos genomas G e H são armazenados em uma matriz binária de $A_{m \times n}$, sendo m = número de genes de G e n = número de genes de H, onde $A_{i,j} = 1$ se e somente se, g_i e h_j formam um match.

A determinação das regiões ortólogas presentes nos genomas envolvidos na comparação é apenas um dos objetivos da Fase 3. A determinação das ROs começa pela determinação dos runs. Um run é representado na matriz binária A por uma diagonal de 1's. Lembrando que apenas as diagonais com pelo menos dois 1's são consideradas um run. A figura 3.2 mostra um exemplo de run encontrado na comparação entre os proteomas dos organismos *Xanthomonas axonopodis* pv.*citri* (CI) e *Xanthomonas campestris* pv. *campestris* (CA) encontrado pela ferramenta EGG.

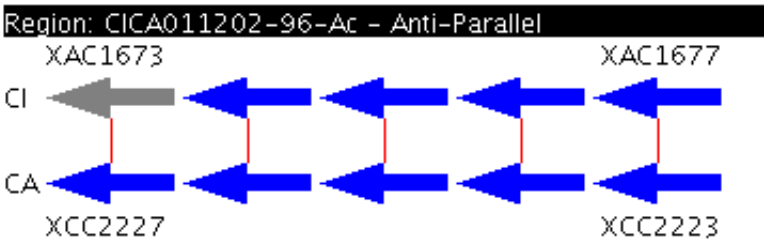


Figura 3.2: run encontradado na comparação entre os organismos *CI* e *CA*

Por fim, EGG encontra as regiões ortólogas existentes entre os dois proteomas através do processo de junção de runs próximos, utilizando como parâmetros

os valores *max_large_gap* e *max_small_gap*. As ROs são determinadas de maneira incremental, no sentido de que uma região resultante da junção de dois runs próximos pode ser ainda juntada com o próximo run à direita. A figura 3.3 mostra uma região ortóloga resultante da junção de dois runs próximos encontrada entre os organismos *Xanthomonas axonopodis* pv. *citri* e *Xanthomonas campestris* pv. *campestris*.

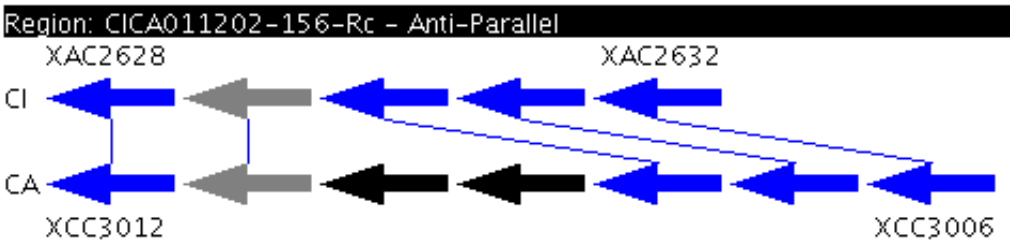


Figura 3.3: RO encontradada na comparação entre os organismos *CI* e *CA*.

3.2 Comparação de Múltiplos Proteomas

A importância da comparação entre múltiplos proteomas está fortemente relacionada ao entendimento do processo de evolução das espécies. Como motivação para a comparação de múltiplos proteomas podemos citar:

- determinação de genes que se preservaram ao longo das espécies;
- determinação de grupos de genes nos quais a ordem e a função foram preservadas;

O principal objetivo do nosso trabalho é a determinação de grupos de genes que preservam a ordem e a funcionalidade entre diversos proteomas distintos, além da disponibilização de uma ferramenta web que permita ao usuário escolher um grupo de organismos a serem comparados, e obtenha como resposta uma representação gráfica das regiões preservadas existentes entre estes organismos.

3.2.1 Regiões Ortólogas Múltiplas

Como dissemos, nosso principal objetivo quando comparamos múltiplos proteomas é a determinação de grupos de genes que se preservaram nos organismos durante o processo de evolução para que possamos obter pistas sobre

a conservação e a funcionalidade destes grupos de genes. A esse grupo de genes preservados chamamos de **região ortóloga múltipla** (ROM). A definição 3.3 apresenta formalmente o conceito de região ortóloga múltipla.

Definição 3.3 *Uma* **região ortóloga mútipla** *R é um conjunto de RGCs de dois ou mais proteomas distintos, tais que, qualquer par de RGCs deste conjunto forma uma região ortóloga.*

Apresentaremos na seção 3.2.2 nossa metodologia para encontrar as ROMs presentes em um conjunto de n proteomas baseado em cliques maximais. Para um melhor entendimento da metodologia proposta, considere o grafo G mostrado na figura 3.4. Note que os vértices v_2, v_3, v_4, e v_5 formam uma clique, pois para qualquer par de vértices (v_i, v_j), $2 \leq i, j \leq 5$ existe uma aresta. A semelhança entre entre a definição de clique e ROM fica evidente, e é por isso que optamos por modelar computacionalmente o nosso problema de comparação múltipla utilizando como estrutura de dados um grafo: para que a determinação das regiões ortólogas múltiplas seja baseada na obtenção das cliques presentes neste grafo. Na figura 3.5 temos a visualização de uma ROM abstrata, e na figura 3.6 apresentamos essa ROM de maneira simplificada, através de uma clique.

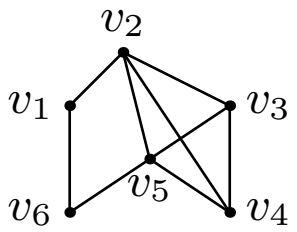


Figura 3.4: Grafo G.

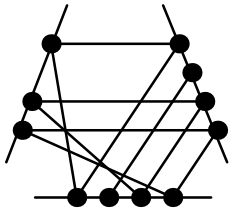


Figura 3.5: ROM abstrada envolvendo 3 proteomas. Cada ponto representa um gene da RCG. As ligações representam matches entre eles.

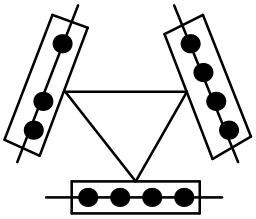


Figura 3.6: ROM simplificada. Cada RGC agora se torna um vértice do grafo, fornecendo uma clique formada por 3 vértices.

3.2.2 Metodologia para determinação das ROMs

A metodologia proposta para determinação das regiões ortólogas múltiplas existentes entre vários proteomas toma como base as comparações dois-a-dois entre todos os pares possíveis de proteomas, utilizando a ferramenta EGG para realizar tais comparações.

O processo de obtenção das cliques pode ser dividido em quatro passos, que serão descritos a seguir.

Passo 1

Este primeiro passo consiste na execução do EGG para cada um dos pares possíveis entre os n proteomas envolvidos na comparação múltipla, para que as regiões ortólogas dois-a-dois sejam encontradas.

Considere o conjunto de n organismos. O número de comparações a serem executadas pelo EGG para esse conjunto é de:

$$\binom{n}{2} = O(n^2) \tag{3.1}$$

Como saída temos as regiões ortólogas existentes para cada par de organismos comparados.

Passo 2

Este passo consiste na construção de um grafo, com base nos resultados obtidos nas comparações dois-a-dois realizadas no passo anterior, que represente as regiões ortólogas existentes entre dois ou mais organismos. A construção do grafo será descrita a seguir.

Cada vértice do grafo será representado por uma RGC que tenha participado de uma RO envolvendo a comparação de dois proteomas. Desta forma, existe uma aresta entre dois vértices no grafo, se e somente se, as RGCs corespondentes formam um RO.

No momento da criação dos vértices, é preciso atentar para o fato de que pode haver sobreposição entre duas RGCs distintas de um mesmo genoma. Neste caso é preciso decidir se estas duas RGCs serão representadas por um ou dois vértices no grafo. Para exemplificar essa situação, considere a figura 3.7, que representa uma RO existente entre os organismos G e H, e a figura 3.8, que representa uma RO entre os organismos G e I.

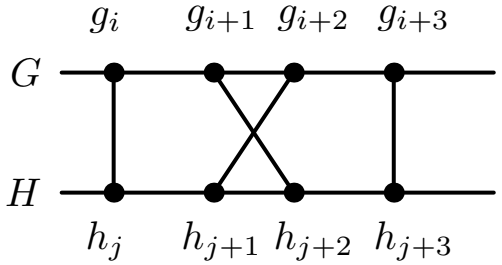


Figura 3.7: RO entre os organismos G e H.

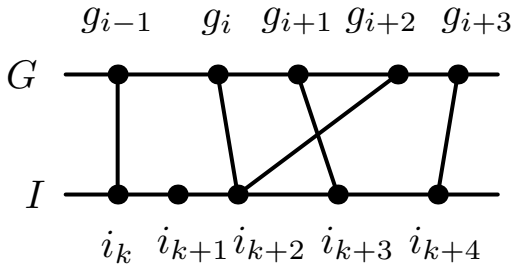


Figura 3.8: RO entre os organismos G e I.

Note que a RGC de G que compõe a RO entre G e H difere apenas do gene g_{i-1} da RGC de G que compõe a RO entre G e I. A criação de dois vértices distintos no grafo para representar as RGCs $(g_i, g_{i+1}, g_{i+2}, g_{i+3})$ e $(g_{i-1}, g_i, g_{i+1}, g_{i+2}, g_{i+3})$ poderia, por exemplo, impedir que uma possível ROM entre os organismo G, H e I fosse descoberta.

Desta forma, adotamos o critério que leva em consideração o tamanho da sobreposição entre as RGCs em número de genes. Assim, caso a sobreposição seja maior ou igual a $P\%$ do tamanho da menor RGC, temos que as duas RGCs serão representadas por um único vértice no grafo, caso contrário, um vértice será criado para cada uma das RGC.

Fica evidente que o número de vértices do grafo irá depender do valor escolhido para o parâmetro P, o que por sua vez irá influenciar no número de ROMs encontradas. No próximo capítulo, mostramos um estudo preliminar sobre a escolha do valor para o parâmetro P.

Ao final deste processo, temos o grafo G representando as relações de ortologia existentes entre os n organismos a serem comparados.

Passo 3

Neste momento, já temos o grafo G representando as relações existentes entre os n genomas envolvidos na comparação, basta encontrarmos as regiões ortólogas múltiplas. Como, pela construção do grafo, temos que uma aresta representa uma relação de ortologia entre dois organismos, nossa estratégia para encontrar regiões ortólogas existentes entre diversos genomas consiste na obtenção de cliques maximais presentes neste grafo. Desta forma, temos que uma clique de tamanho k representa uma ROM existente entre k genomas distintos.

Antes de apresentarmos o algoritmo utilizado para obtenção de cliques maximais, faremos algumas considerações.

O problema da clique máxima consiste em encontrar a clique com maior número de vértices existente em um grafo. Esse problema é NP-Difícil, logo não se conhece algoritmo polinomial para solucioná-lo [5]. Um algoritmo que encontra todas as cliques maximais de um grafo G, encontra também a clique máxima, logo, uma solução em tempo polinomial não possível.

Um algoritmo exato para obtenção de todas as cliques maximais foi implementado, porém os testes realizados mostraram que, mesmo para um conjunto pequeno de organismos a serem comparados, o tempo de execução do

algoritmo pode ser elevado, o que inviabilizaria a execução *online* das comparações.

Devido a esse fato, optamos pela implementação de uma heurística para a solução do problema de cliques maximais, a qual não garante que todas as cliques maximais sejam encontradas, porém resultados biologicamente relevantes foram observados, como por exemplo, regiões envolvendo operons conhecidos de organismos.

A heurística implementada para obtenção de cliques maximais é bastante simples. Inicialmente cada vértice é considerado uma clique. Para cada umas das cliques iniciais, trabalha-se na tentativa de expandí-la através da inserção de novos vértices. Esse processo continua até que a expansão não seja mais possível, ou seja, no momento em que a inserção de um novo vértice faz com que o conjunto não represente mais uma clique o processo termina e uma nova clique acaba de ser encontrada. Ao final, cliques repetidas ou contidas em outras são eliminadas.

Uma ordem para as tentativas de inserções de vértices em uma clique é necessária. A heurística implementada considera os vértices na ordem crescente. A figura 3.9 e figura 3.10 mostram duas possíveis ordenações para os vértices do grafo G.

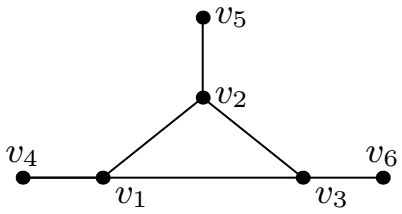


Figura 3.9: Primeira ordenação para os vértices de G.

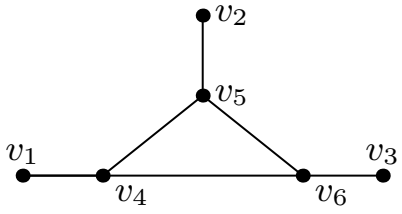


Figura 3.10: Segunda ordenação para os vértices de G.

Considerando a ordenação dos vértices mostrada na figura 3.9, as cliques $C_1 = \{v_1, v_2, v_3\}$, $C_2 = \{v_4, v_1\}$, $C_3 = \{v_5, v_2\}$ e $C_4 = \{v_6, v_3\}$ seriam encontradas. Para a ordenação mostrada na figura 3.10, seriam encontradas apenas as cliques $C_1 = \{v_1, v_4\}$, $C_3 = \{v_2, v_5\}$ e $C_4 = \{v_3, v_6\}$.

Na figura 3.11 temos a descrição do algoritmo utilizado para encontrar clique maximais em um grafo G com n vértices e m arestas.

Início Algoritmo

1. inicialize cada uma das n cliques iniciais com um vértice distinto do grafo
2. **para** cada uma das n cliques iniciais **faça**

   Verifique, em ordem crescente, a possibilidade de inserção de cada um dos vértices do grafo no conjunto, sem que este deixe de ser uma clique.
3. **para** cada clique C encontrada **faça**

   Verifique no conjunto de cliques se existe alguma outra clique que a contenha. Caso isso ocorra, elimine a clique C.

Fim Algoritmo

Figura 3.11: Algoritmo para encontrar cliques maximais presentes em um grafo G.

Ao final deste capítulo, são feitas considerações sobre a complexidade desse algoritmo.

Passo 4

Obtidas as cliques, ou seja, as **ROM**s, é preciso representá-la de maneira adequada para que as relações de homologia fiquem aparentes. A forma escolhida para a visualização das ROMs é através da construção de um Alinhamento Múltiplo entre as RGCs pertencentes a ROM.

O algoritmo para a visualização das ROMs é uma modificação do algoritmo do Gusfield [14] descrito no capítulo 2. A principal diferença do nosso algoritmo é que não utilizamos nenhuma medida de pontuação para determinarmos a seqüência centro para a construção do alinhamento, o que o torna mais rápido. Quem decide qual será o organismo centro do alinhamento, o qual chamaremos de âncora e denotamos por s_c, é o usuário, pois desta forma nosso programa consegue responder a perguntas como:

- quais os genes do organismo s_c se preservaram durante o processo de evolução;
- quais os genes de s_c não se preservaram;
- para uma RGC conhecida de s_c, quais os outros organismos que possuem RGCs que formam um RO com a RGC de s_c.

Lembre-se que nosso interesse é a construção de um alinhamento múltiplo que represente as relações de ortologia entre RGCs de genomas distintos, ou seja, estaremos construindo alinhamento entre genes. Em seguida, descrevemos o algoritmo desenvolvido para a construção do alinhamento múltiplo entre proteomas.

Simplificadamente o funcionamento do algoritmo é o seguinte: para uma ROM constituída por k RGCs de genomas distintos, uma delas é tomada como âncora para a construção do alinhamento e as demais RGCs do conjunto são progressivamente inseridas neste alinhamento, considerando apenas o alinhamento desta nova RGC com a RGC âncora. Vamos considerar que uma RGC_i, para $1 \leq i \leq k$, representa uma RGC pertencente ao organismo G_i e que está presente em uma ROM de k organismos, digamos $G_1, G_2, ..., G_k$. Na figura 3.12 descrevemos formalmente o algoritmo que constrói o alinhamento múltiplo (AM) para uma dada ROM de tamanho k.

Faremos algumas considerações à respeito da representação do alinhamento entre RGCs, no que diz respeito a forma como um gene é visualizado neste alinhamento.

Como dissemos, cada gene pertence a uma das fitas do DNA, fita "+" ou fita "-". Em nossa representação, um gene da fita "+" é representado por uma figura de seta direcionada para a direita, enquando que um gene na fita "-" é representado por uma seta direcionada para a esquerda. Vimos também, na seção 3, que um par de ortólogos é formado por dois genes que formam um *match* ou por dois genes que formam um BBH, desta forma, um gene pode assumir a cor vermelha ou azul, dependendo se este gene forma um *match* ou um BBH com o gene da âncora, respectivamente.

Considere o alinhamento que representa uma ROM encontrada entre n organismos. Seja este alinhamento armazenado em uma tabela M de dimensões n linhas e l colunas. Para uma dada coluna m do alinhamento, podemos garantir apenas a existência do par ortólogo formado entre os genes armazenados nas posições $M_{0,m}$ e $M_{i,m}$, para $1 \leq i < n$ e $0 \leq m < l$, pelo fato de que o alinhamento foi construído com base apenas na âncora. Porém, há uma importante consideração a ser feita em relação aos demais pares de

Início Algoritmo

1. inicialize a região âncora do alinhamento

   $R_c \leftarrow RGC_i$, escolhida pelo usuário

2. **para** cada RGC_j presente na ROM e que ainda não foi inserida no alinhamento R **faça**

   adicione RGC_j ao alinhamento da seguinte forma: encontre o alinhamento entre RGC_j e R_c, obtendo RGC'_j e R'_c. Em seguida ajuste as seqüências já alinhadas inserindo espaços nas colunas onde espaços foram adicionados em R_c para obtenção de R'_c

   $R_c \leftarrow R'_c$

Fim Algoritmo

Figura 3.12: Algoritmo para a construção do AM de uma ROM.

genes possíveis para uma coluna m do alinhamento. Como uma ROM pode ser vista como uma clique de RGCs, temos que todas as RGCs, duas-a-duas, formam uma região ortóloga, e as chances de que essa relação fique aparente no alinhamento múltiplo construído com base apenas em uma RGC âncora é grande, quanto tratamos de organismos próximos filogeneticamente.

A figura 3.13 mostra um exemplo de **ROM** encontrada pela nossa ferramenta, envolvendo os organismos *Escherichia coli K-12* (ec), *Chlamydia trachomatis* (ch) e *Bacillus halodurans C-125* (bh), na qual para qualquer par de genes de uma coluna fixa k do alinhamento, temos um par de genes ortólogos. Outros exemplos de regiões ortólogas múltiplas serão apresentados mais adiante.

Complexidade

Faremos agora um estudo sobre a complexidade dos algoritmos para a obtenção de cliques maximais e do alinhamento múltiplo entre as RGCs de uma ROM.

Considere o grafo G com m arestas e n vértices. A obtenção das cliques iniciais consome tempo linear $O(n)$, já que inicialmente cada vértice é consi-

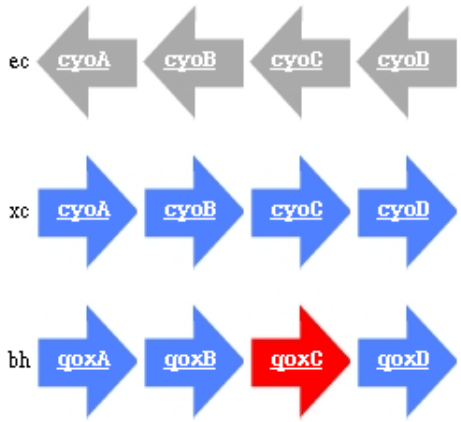


Figura 3.13: Exemplo de uma ROM. As nomenclaturas utilizadas para os organismos encontra-se descrita na tabela 4.1.

derado uma clique. O tempo gasto para a construção de uma clique é $O(n^2)$, uma vez que todos os vértices do grafo são candidatos a participar desta clique, e por isso devem ser testados. Como temos inicialmente n cliques, o custo total para encontrar todas elas é $O(n^3)$. A eliminação das cliques não maximais é um processo computacionalmente caro, uma vez que para cada uma das cliques é necessário verificar se há intersecção desta com todas as outras cliques encontradas. Porém, no nosso problema, o número de vértices de uma clique é pequeno, se comparado com o número de vértices do grafo. Considerando que o tamanho máximo entre todas as cliques encontradas é k, o custo do processo de eliminação das cliques não maximais é $O(n^2k^2)$. Desta forma, o custo do algoritmo é $O(n^3)$.

Para o algoritmo de alinhamento múltiplo é importante lembrar que cada nova região a ser inserida no alinhamento só será alinhada com a região âncora, e que todos os alinhamentos dois-a-dois já foram encontrados.

O passo inicial do algoritmo tem custo $O(1)$, e consiste na escolha da RGC âncora para o alinhamento. Seja l o número máximo de colunas da RGC âncora no alinhamento e k o número de RGCs a serem alinhadas. A inserção da k-ésima nova RGCs no alinhamento múltiplo tem o custo correspondente ao número de colunas da RGC âncora no alinhamento, ou seja $O(l)$. Após a inserção da k-ésima RGC no alinhamento, a inserção de espaços nas RGCs já alinhadas pode ser necessário para manter a consistência do alinhamento. O custo desta operação para cada seqüência inserida é $O((k-1)l)$. Desta forma, o custo total do algoritmo é $kl + k((k-1)l)$, ou seja $O(k^2)$.

Capítulo 4

BAGRE

A ferramenta resultante de nosso trabalho denomina-se BAGRE - *Base de Dados de Genes, Genomas e Regiões Ortólogas* e está disponível em uma página web no endereço *http://bagre.dct.ufms.br/~montera/mestrado*. A seguir na seção 4.1, apresentamos o funcionamento da ferramenta BAGRE de uma forma geral. Na seção 4.2 é descrito o banco de dados criado para um melhor desempenho do sistema. Na seção 4.3 são apresentados detalhes de implementação de todas as ferramentas utilizadas no decorrer do trabalho, como o servidor web utilizado, o sistema operacional e as linguagens de programação. Alguns exemplos de regiões ortólogas múltiplas encontradas envolvendo alguns dos organismos presentes na tabela 4.1 serão apresentados na seção 4.4. Por fim, na seção 4.5 apresentamos uma breve descrição de outros trabalhos relacionados ao tema estudado.

4.1 Descrição Geral da Ferramenta

Como dissemos, nossa ferramenta encontra-se disponível por meio de um site na web e é de fácil utilização. A figura 4.1 mostra a tela inicial da ferramenta, onde o usuário escolhe os organismos a serem comparados e o tipo de comparação a ser realizada.

A obtenção das ROMs existentes entre os organismos depende dos valores de determinados parâmetros. Estes parâmetros podem ser informados pelo usuário e são eles:

Figura 4.1: Tela inicial da ferramenta.

- número mínimo de organismos envolvidos em cada ROM a ser buscada;
- sobreposição mínima exigida entre as RGC candidatas a formar um mesmo vértice no grafo;
- visualizar somentes genes que formam BBH com a âncora;
- organismo âncora para a construção do alinhamento.

Caso o usuário não informe alguns destes parâmetros, os valores 2, 90, "não" e "qualquer" são utilizados como valores *default*, representando repectiva-

mente, o número mínimo de organismos envolvidos em uma ROM, a sobreposição mínima entre as RGCs, visualização somente de genes que formam BBH com a âncora e o organismo âncora para a construção do alinhamento. A figura 4.2 mostra a tela onde o usuário pode ajustar os parâmetros para comparação.

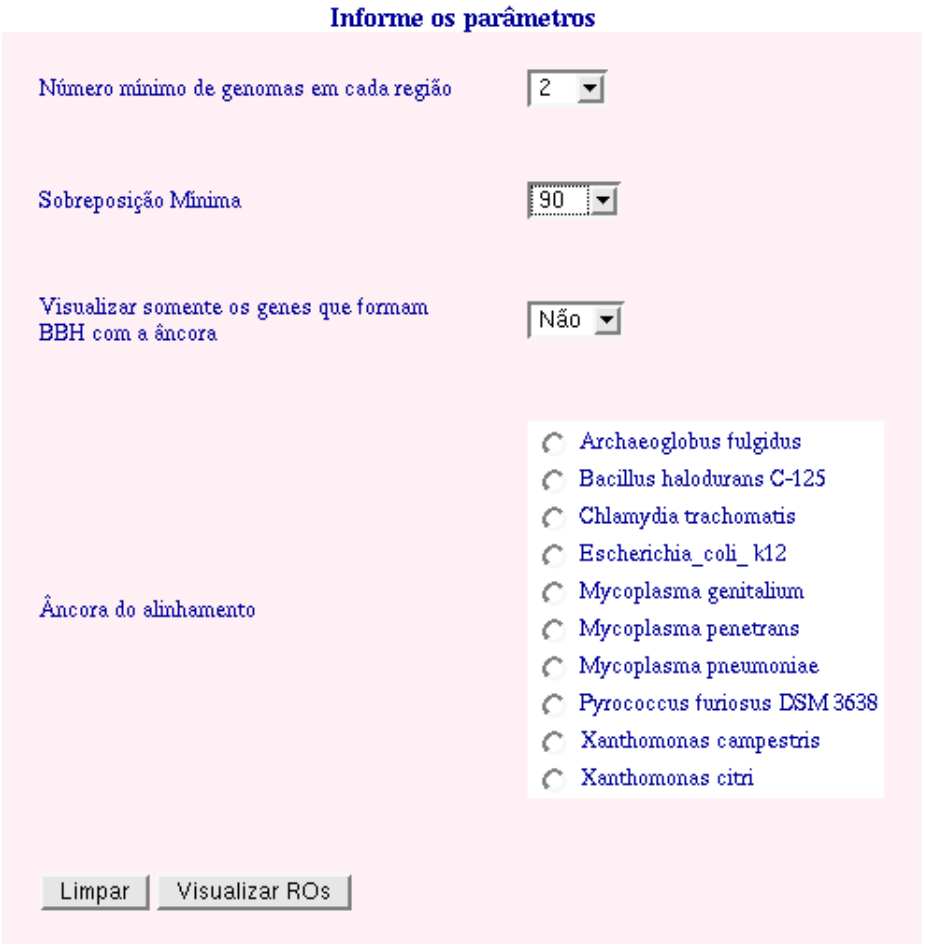


Figura 4.2: Tela para informação dos parâmetros da comparação.

Faremos neste momento, algumas considerações sobre o parâmetro P.

Sejam duas RGCs R_1 e R_2 de um mesmo genoma G que se sobrepoem. Como foi visto na seção 3.2.2 do capítulo 3, é preciso calcular o valor da sobreposição da menor RGC em relação a maior para decidir se estas RGCs darão origem a dois ou apenas um vértice no grafo. Para justificar a escolha da menor RGC para o cálculo da sobreposição, considere a figura 4.3.

Note que temos 40% dos genes de R_1 contidos em R_2 e 80% dos genes de

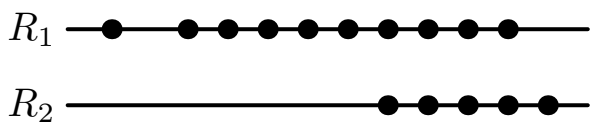


Figura 4.3: Sobreposição entre R_1 e R_2.

R_2 contidos em R_1. Se a exigência para o cálculo da sobreposição fosse da maior RGC em relação a menor, para um valor qualquer de P $>$ 40, as duas regiões não poderiam ser representadas por um único vértice no grafo. Tal situação poderá impedir que uma clique seja formada no grafo, impedindo assim que uma ROM seja encontrada.

Devido a esta exigência, na maioria dos casos, o valor escolhido para o parâmetro P não influencia no número de vértices do grafo, pois geralmente ocorre que a menor região encontra-se inteiramente contida na maior.

A escolha da âncora não influencia na obtenção das ROMs, porém ela é uma escolha muito importante para a visualização destas. Considere como exemplo a ROM que envolve os organismos *Aquefix aeolicus* (aa), *Escherichia coli* k12 (ec), *Chlamydia trachomatis* (ch), *Borrelia burgdorferi* (bb), *Archaeoglobus fulgidus* (af), apresentada na figura 4.4. Caso o organismo af fosse tomado como âncora, a visualização desta mesma ROM seria comprometida, pois este organismo possui poucos genes em sua RGC, o que deixaria o alinhamento com muitas colunas vazias.

Informações adicionais sobre os genes envolvidos em uma ROM podem ser obtidas através de um clique no elemento gráfico que o representa. As informações são apresentadas em uma página html e são elas: posição de início e fim da seqüência de aminoácidos que compõem o gene no genoma; fita a qual o gene pertence; PID; Gene; sinônimo; produto; seqüência de aninoácidos que compõem o gene além de um link para a página do $NCBI$ (*www.ncbi.nih.gov*), onde pode-se encontrar informações como autores e local de publicação do gene, entre outros.

A figura 4.5 mostra um exemplo das informações adicionais para o gene $rl5$ pertencente ao organismo $Chlamydia\ trachomatis$.

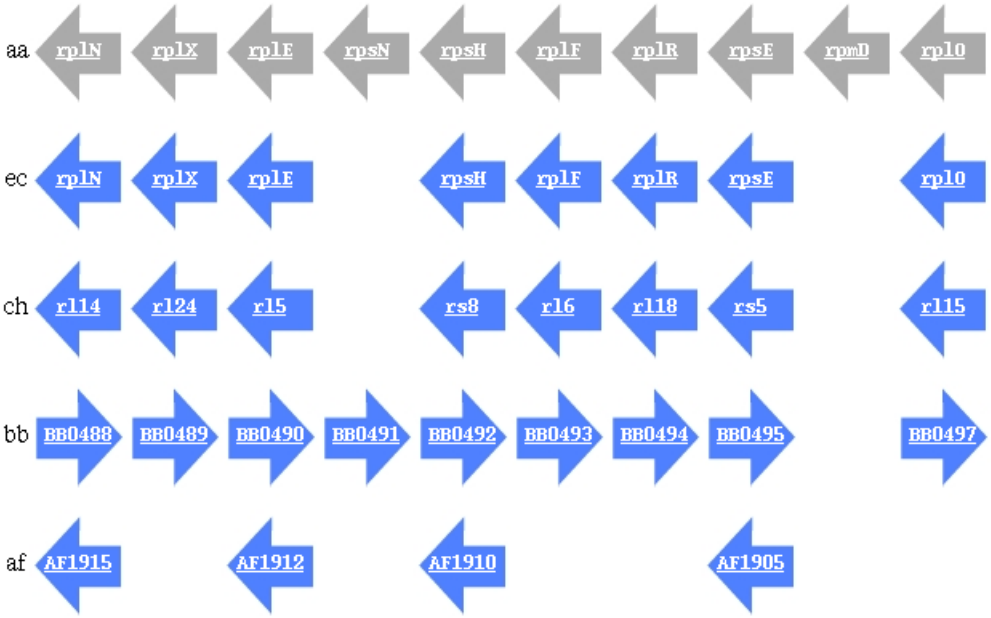


Figura 4.4: ROM envolvendo 5 organismos. As nomenclaturas utilizadas para os organismos encontra-se descrita na tabela 4.1.

Location	Strain	PID	Gene	Synonym	Product
590272..590814	-	3328953	rl5	CT516	L5 Ribosomal Protein

```
ORIGIN
 >gi|3328953|gb|AAC68117.1| L5 Ribosomal Protein [Chlamydia trachomatis] MSRLKKL
  YTEEIRKTLQDKFQYENVMQIPVLKKIVISMGLAEAAKDKNLFQAHLEELAVISSQKPLVTRARNSIAGFKLREGQGI
  GAKVTLRGIRMYDFMDRFCNIVSPRIRDFRGFSCKGDGRGCYSFGLDDQQIFPEVDLDRVKRSQGMNITWVTTAQTDA
  ECLTLLECMGLRFKKAQ
```

Click here to see genbank record

Figura 4.5: Informações adicionais sobre o gene $rl5$.

4.2 O Banco de Dados

Para dar suporte aos programas executados por meio da página web, decidimos pela criação de um banco de dados para realizar o armazenamento e a recuperação das informações necessárias às comparações. Como, para

cada execução da ferramenta de comparação, o conjunto de entrada pode variar em quantidade e organismos, a manutenção estática dessas informações torna-se inviável.

O Sistema Gerenciador de Banco de Dados escolhido para a criação e manutenção do base de dados foi o MySql. O MySql é um servidor robusto de bancos de dados SQL (*Structured Query Language*), além de ser um software de utilização livre.

O banco de dados desenvolvido é composto por 2 tabelas: a tabela de `genomas` e a tabela de comparações dois-a-dois, chamada `pairwise`. Descreveremos a seguir essas duas tabelas.

Na tabela de `genomas`, são mantidas as seguintes informações:

- `NOME`: string que armazena o nome do organismo;
- `ID`: string que armazena uma identificação para o organismo;
- `GENES`: inteiro que armazena o número de genes existentes no organismo;
- `BASES`: inteiro que informa o número de pares de bases do organismo;
- `ACESSO`: string que identifica o arquivo que contém informações adicionais sobre o organismo.

Dado um conjunto de organismos, os quais foram selecionados para a realização da comparação múltipla, tais informações são disponibilizadas ao usuário antes que a comparação seja feita. Através do campo `ACESSO` de um organismos podemos disponibilizar ainda o arquivo de extensão ".ptt", o qual traz informações sobre cada um dos genes existentes no organismo. A figura 4.6 mostra um exemplo desse arquivo para o organismo $Aquifex\ aeolicus$.

Na tabela `pairwise` estão armazenadas as informações referentes às comparações dois-a-dois entre todos os organismos disponíveis para a realização da comparação múltipla. As informações armazenadas são:

- `ID_G1`: string de identificação do genoma 1 envolvido na comparação;
- `ID_G2`: string de identificação do genoma 2 envolvido na comparação;
- `MATCHES`: inteiro que armazena o número de matches existentes entre os genes do genoma 1 e do genoma 2;

- `BBHS`: inteiro que armazena o número de *Bidirectional Best Hits* existentes entre os genes do genoma 1 e do genoma 2;
- `ROs`: inteiro que armazena o número de regiões ortólogas encontradas;
- `ESP_G1`: inteiro que armazena o número de genes específicos do genoma 1;
- `ESP_G2`: inteiro que armazena o número de genes específicos do genoma 2.

```
Aquifex aeolicus complete genome - 0..1551335
1522 proteins

       Location Strand  Length PID Gene Synonym Code COG Product
          1..2100  +  699 2982777 fusA aq_001 J COG0480 elongation factor EF-G
       2117..3334  +  405 2982777 tufA1 aq_005 J COG0050 elongation factor EF-Tu
       3346..3660  +  104 2982773 rpsJ aq_008 J COG0051 ribosomal protein S10
       3665..4390  +  241 2982767 rplC aq_009 J COG0087 ribosomal protein L03
       4387..4986  +  199 2982768 rplD aq_011 J COG0088 ribosomal protein L04
       4990..5301  +  103 2982771 rplW aq_012 J COG0089 ribosomal protein L23
       5313..6227  +  304 2982766 rplB aq_013 J COG0090 ribosomal protein L02
       6340..6900  +  186 2982775 rpsS aq_015 J COG0185 ribosomal protein S19
       7018..7314  +  98 2982770 rplV aq_016a J COG0091 ribosomal protein L22
       7317..7955  +  212 2982772 rpsC aq_017 J COG0092 ribosomal protein S03
       8017..8445  +  142 2982769 rplP aq_018 J COG0197 ribosomal protein L16
       8678..9001  +  107 2982774 rpsQ aq_020 J COG0186 ribosomal protein S17
       9001..9660  +  219 2982763 aroD aq_021 E COG0710 3-dehydroquinate dehydratase
       9657..10157  - 166 2982778 aq_022 aq_022 R COG1266 putative protein
      10169..11299  - 376 2982764 argD aq_023 E COG0160 N-acetylornithine aminotransferase
      11296..12609  - 437 2982765 nsd aq_024 M COG1004 nucleotide sugar dehydrogenase
      12887..14005  - 372 2982781 rodA aq_025 D COG0772 rod shape determining protein RodA
      14174..14656  - 160 2982785 aq_026 aq_026 I COG3255 putative protein
      14676..16454  - 592 2982789 aq_027 aq_027 T COG2200 hypothetical protein
      16527..17006  - 159 2982786 aq_028 aq_028 --putative protein
      17053..18261  +  402 2982782 moeA1 aq_030 H COG0303 molybdenum cofactor biosynthesis protein A
      18275..19615  +  446 2982784 trnS aq_031 P COG0471 transporter (Pho87 family)
      19620..21062  +  480 2982790 aq_032 aq_032 S COG1690 hypothetical protein
      21051..22346  - 431 2982791 aq_035 aq_035 T COG2199 hypothetical protein
      22379..22885  +  168 2982792 aq_036 aq_036 H COG1057 hypothetical protein
      22998..23525  +  175 2982787 aq_037 aq_037 --putative protein
      23541..24407  +  288 2982783 hibD aq_038 I COG2084 3-hydroxyisobutyrate dehydrogenase
      24404..24982  - 192 2982780 hisB aq_039 E COG0131 imidazoleglycerolphosphate dehydratase
      24997..26217  - 406 2982788 aq_040 aq_040 R COG0517 putative protein
      26274..26585  - 103 2982805 aq_041 aq_041 L COG0792 putative protein
      26618..27340  - 240 2982798 cyc aq_042 C COG2857 cytochrome c
      ...
```

Figura 4.6: Arquivo ".ptt".

4.3 Detalhes de Implementação

A configuração da máquina utilizada foi: Intel Pentium 4 de 1.7 GHz com 512 MB de memória RAM, 256 KB de memória cache e sistema operacional Linux Red Hat 9.0. O servidor Web escolhido foi o Apache 2.0.40-21.3 com suporte a PHP, e o Sistema Gerenciador de Banco de Dados o MySql.

Os programas desenvolvidos foram escritos na liguagem C e o compilador utilizado foi o GCC (GNU software). Antes de descrevermos estes programas, descreveremos um dos arquivos de saída gerado pelo programa EGG como resultado da comparação entre dois proteomas, pelo fato de fazermos uso deste arquivo. Considere o seguinte trecho do arquivo, chamado *chbb.mul* produzido pelo EGG como resultado da comparação entre os organismos $Chlamydiatrachomatis$ e $Borreliaburgdorferi$.

1. ch 1042519
2. bb 910724
3. 7
4. CHBB020605-1-Rc
5. 26 29
6. 693 698
7. 4
8. 29 698 1
9. 28 697 1
10. 27 694 1
11. 26 693 1
12. CHBB020605-2-Ac
13. 96 99
14. 799 802
15. 4
16. 99 799 1
17. 98 800 1
18. 97 801 1
19. 96 802 1
20. ...

Nas linhas 1 e 2 temos a sigla de identificação e o número de pares de bases para o genoma dos organismos *Chlamydia trachomatis* e *Borrelia* $burgdorferi$, respectivamente; o valor da linha 3 indica o número de regiões ortólogas encontradas entre os dois organismos, no exemplo 7; nas linhas seguintes temos as informações referentes a cada uma das $k = 7$ ROs existentes entre os organismos *ch* e *bb*. Para cada uma das ROs temos: uma string indentificadora da região; gene inicial e final da RCG no organismo *ch*; gene inicial e final da RCG no organismo *bb*; número de *matchs* entre os genes da RO; para cada um dos *matchs* temos: gene no organismo *ch*; gene no organismo *bb*; e o valor 1 ou 0, dependendo se os genes formam BBH ou não, respectivamente.

A seguir, serão descritos os programas desenvolvidos em nosso trabalho.

O primeiro programa é chamado *grafo.c*. Este programa tem como entrada os arquivos "*.mul*" resultantes das comparações dois-a-dois realizadas pelo programa EGG, entre todos os organismos envolvidos na comparação múltipla, e modela tais informações em um grafo, segundo a medodologia descrita no capítulo 3. O grafo resultante é armazenado em um arquivo de saída chamado *grafo.txt*.

O segundo programa a ser executado é chamado *clique.c*, que toma como entrada o grafo gerado pelo programa *grafo.c* e determina todas as cliques maximais presentes neste grafo.

Por fim, o último programa a ser executado é o programa *regioes.c*, o qual produz o alinhamento múltiplo que representa cada uma das cliques maximais encontradas no grafo, ou seja, produz o alinhamento múltiplo entre as RGCs que formam cada uma das ROMs existentes entre os organismos comparados. Note que, pelo arquivo de entrada "clique.c", obtemos as ROMs existentes entre os organismos apenas em termos dos números seqüenciais dos genes no genoma de origem. Informações como, a ordem do gene no genoma, a fita, o identificador e produto anotado para cada um dos genes de um genoma são mantidas no banco de dados, como vimos na seção 4.2, consultas são realizadas e estas informações obtidas e utilizadas para uma melhor visualização e interpretação das regiões ortólogas múltiplas encontradas.

A necessidade de fazer com que a ferramenta de comparação desenvolvida estivesse disponível na web nos levou a utilização do protocolo de comunicação chamado **CGI**. O termo CGI vem do inglês **C***ommon* **G***ateway* **I***nterface* e é um protocolo de comunicação através do qual um servidor web intermedia a transferência de informações entre um programa (que se encontre no mesmo computador que o servidor web), no nosso caso programas escritos na linguagem C, e um cliente HTTP.

4.4 Resultados

Nesta seção serão mostrados alguns exemplos de regiões ortólogas múltiplas envolvendo alguns subconjuntos de genomas, dentre os mostrados na tabela 4.1, encontrados pela ferramenta desenvolvida.

Sejam os organismos *Borrelia burgdorferi* (bb), *Escherichia coli* k12 (ec) e *Synechocystis PCC6803* (sp). Inicialmente as comparações dois-a-dois entre cada par possível de organismos foram realizadas. Os resultados obtidos nesta comparação são exatamente os mesmos fornecidos pelo EGG, porém visualizados de uma forma diferente. As figuras 4.7, 4.8 e 4.9 mostram exemplos de

uma RO encontrada entre os organismos *Escherichia coli* k12 e *Synechocystis PCC6803*, *Escherichia coli* k12 e *Borrelia burgdorferi* e entre os organismos *Synechocystis PCC6803* e *Borrelia burgdorferi*, respectivamente.

Organismo	**Sigla**	**#bp**
Aeropyrum pernix K1	ap	1669695
Aquefix aeolicus	aa	1551335
Archaeoglobus fulgidus	af	2178400
Bacillus halodurans C-125	bh	4202353
Borrelia burgdorferi	bb	910724
Chlamydia trachomatis	ch	1042519
Escherichia_coli_ k12	ec	4639221
Heamuphilus_influenzae	hi	1830138
Helicobacter pyroli_26695	hp	1667867
Methanobacterium thermoautotrophicum	mh	1751377
Methanococcus jannaschii	mj	1664970
MycobacteriumtuberculosisH37Rv (sanger)	mt	4411529
Mycoplasma genitalium	mg	80074
Mycoplasma penetrans	mn	1358633
Mycoplasma pneumoniae	mp	816394
Mycobacterium leprea strain TN	ml	3268203
Pseudomonas aeruginosa PA01	pa	6264403
Methanobacterium thermoautotrophicum	mh	1751377
Methanococcus jannaschii	mj	1664970
Pseudomonas aeruginosa PA01	pa	6264403
Pseudomonas putida KT2440	pp	6181863
Pseudomonas syringae pv. tomato str. DC3000	ps	6397126
Pyrococcus abyssi	pb	1765118
Pyrococcus furiosus DSM 3638	pf	1908256
Pyrococcus horikoshii	ph	1738505
Synechocystis PCC6803	sp	3573470
Trepomena pallidum	tp	1138011
Xanthomonas campestris	xp	5076188
Xanthomonas citri	xc	5175554

Tabela 4.1: Organismos disponíveis para comparação.

Observando as três ROs mostradas nas figuras 4.7, 4.8 e 4.9, é possível notar que existe uma ROM entre estes três organismos. Executamos então a comparação simultânea dos três organismos e encontramos, entre outras, a ROM esperada, a qual é mostrada na figura 4.10.

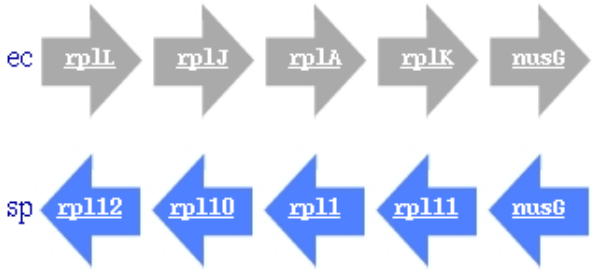


Figura 4.7: RO envolvendo os organismos ec e sp.

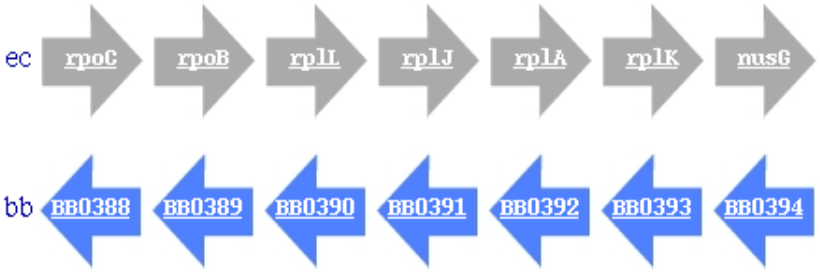


Figura 4.8: RO envolvendo os organismos ec e bb.

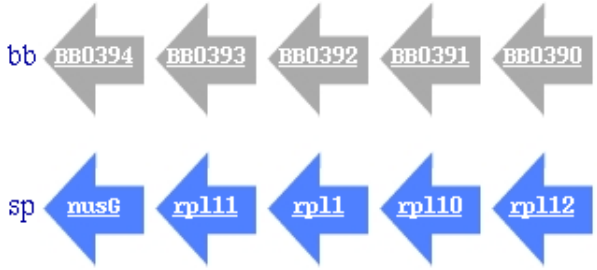


Figura 4.9: ROM envolvendo os organismos sp e bb.

A seguir é apresentado um exemplo de como a escolha da âncora pode influenciar na visualização de uma determinada ROM. Seja uma região ortóloga múltipla encontrada entre os organismos *Escherichia coli* k12 (ec), *Chlamydia trachomatis* (ch), *Trepomena pallidum* (tp) e *Borrelia burgdorferi* (bb). A figura 4.11 mostra uma ROM encontrada envolvendo os quatros organismos citados anteriormente cuja âncora é o organismo *Borrelia burgdorferi*. Se o organismo escolhido como âncora fosse o organismo *Chlamydiatrachomatis* a ROM seria visualizada como mostrada na figura 4.12.

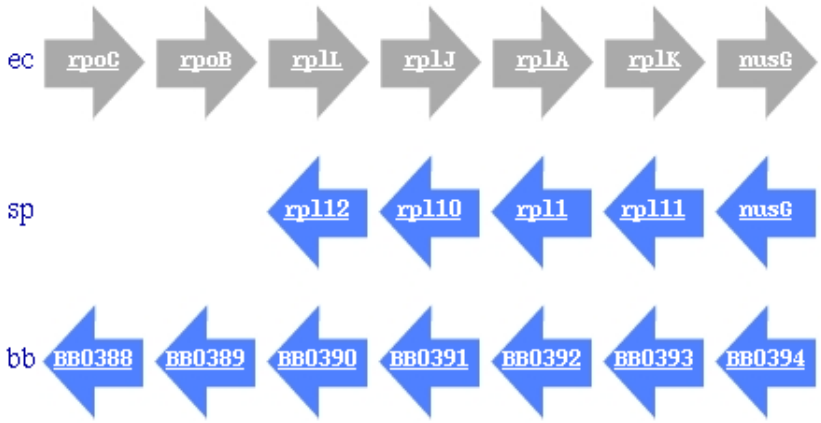


Figura 4.10: ROM envolvendo os organismos ec, sp e bb.

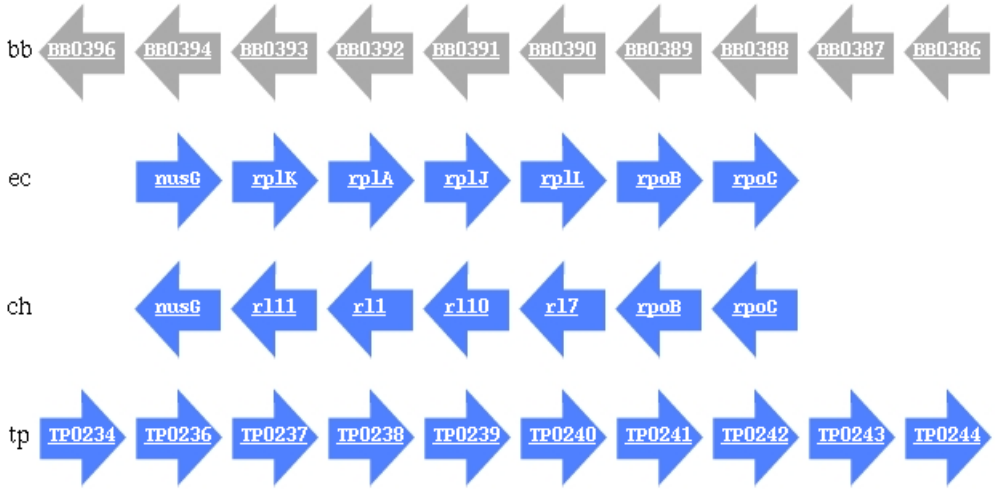


Figura 4.11: ROM envolvendo os organismos bb, ec, ch e tp.

A figura 4.13 apresenta uma ROM bem comportada. Uma ROM é dita ser bem comportada se para qualquer par de genes de uma mesma coluna tem-se um par ortólogo.

A figura 4.14 mostra um conhecido operon do organismo *Escherichia coli* k12 o qual formou uma região ortóloga com o organismo *Aquefix aeolicus*.

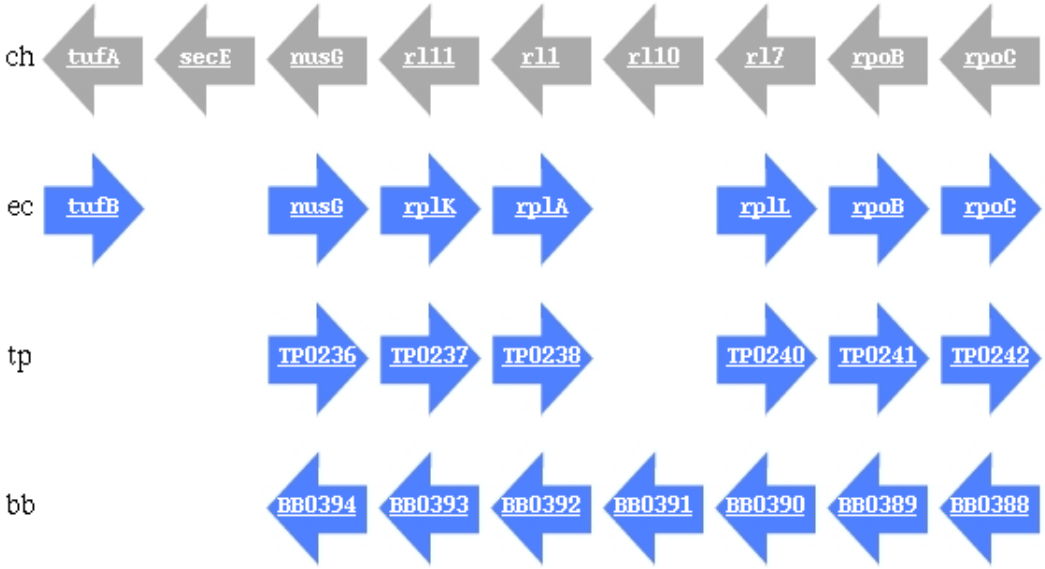


Figura 4.12: ROM envolvendo os organismos ch, ec, tp e bb.

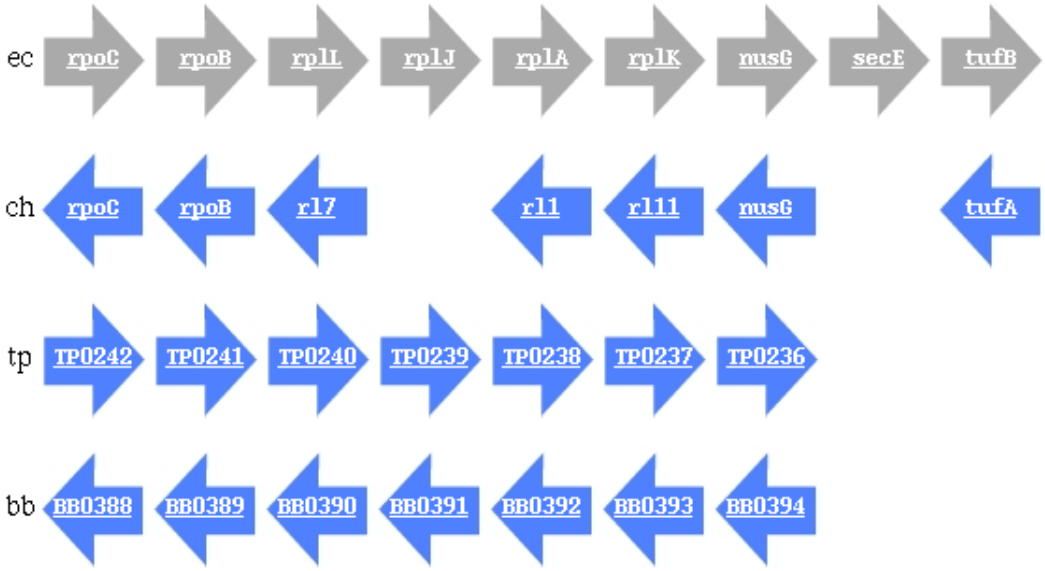


Figura 4.13: ROM bem comportada.

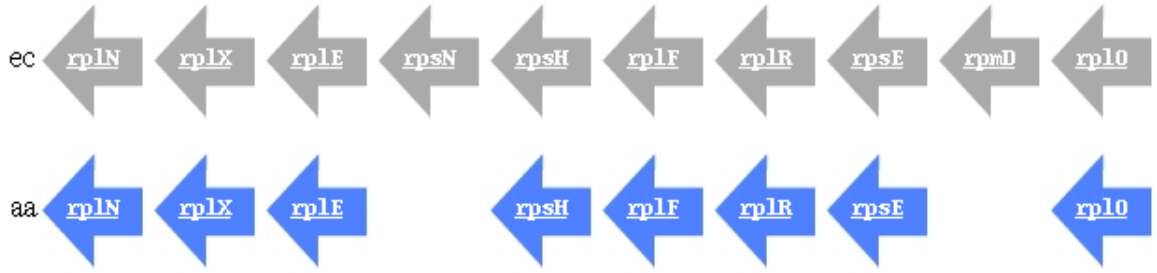


Figura 4.14: Operon da ec conservado no organismo aa.

4.5 Outros Trabalhos

Faremos neste momento, breves comentários sobre alguns trabalhos existentes e relacionados ao tema de nossa pesquisa, os quais nos auxiliaram no desenvolvimento da mesma.

Tamames [35] em seu trabalho analisa características da conservação da ordem em genomas de procariotos e conclui que, quanto menor a distância filogenética entre os organismos, maior é a conservação da ordem dos genes; e que, à medida em que a distância filogenética aumenta, a conservação da ordem dos genes tende a ser perdida. Para a comparação entre os organismos envolvidos na pesquisa foi utilizada a ferramenta BLAST [4].

Fujibuchi e outros [12] em associação Ogata e outros [24] desenvolveram ferramentas para a detecção automática de clusters de genes em múltiplos genomas, baseado na comparação de grafos. Primeiramente, clusters de genes são encontrados (regiões ortólogas) [24] a partir da comparação dois-a-dois dos genomas envolvidos, utilizando o programa SSEARCH, o qual é uma implementação para o algoritmo de Smith-Waterman [32]. Posteriormente, um algoritmo de "clusterização" é aplicado para encontrar os clusters em múltiplos genomas, seguindo o método P-quasi de ligação.

KEGG, *Kyoto Encyclopedia of Genes and Genomes*, é uma base de dados envolvendo genes, genomas e vias metabólicas, desenvolvido por Kanehisa e Goto [17], a qual tem como base os trabalhos de Fujibuchi e outros [12, 24] e Ogata e outros [24].

STRING [33] é um web-server que encontra, para um dado gene de entrada, chamado de "query", organismos que possuem clusters contendo esse gene. Tal ferramenta oferece uma visualização dos resultados encontrados, que permite analisar toda uma vizinhança dos organismos que contém o gene query.

As principais diferenças entre nossa pesquisa e os trabalhos citados acima estão relacionadas às metodologias usadas na determinação das regiões ortólogas (das comparações dois-a-dois), assim como na determinação dos clusters (ROM). Além disso, estamos interessados em tratar apenas de genomas de procariotos, proporcionando um conjunto mais detalhado de informações, já que genomas de procariotos são mais próximos filogeneticamente entre si e, por conseqüência, tendem a preservar a ordem e funcionalidade dos genes.

Capítulo 5

Conclusões e Perspectivas

Neste trabalho propomos uma metodologia baseada na obtenção de cliques maximais em grafos, para encontrar regiões de conteúdo gênico conservado em múltiplos genomas. Tais regiões, uma vez encontradas, podem servir como pistas para investigação envolvendo conjunto de genes relacionados a funcionalidades importantes, como por exemplo, operons de procariotos.

As regiões ortólogas múltiplas são determinadas a partir das regiões encontradas nas comparações dois-a-dois dos genomas. Essas regiões dois-a-dois são determinadas pela ferramenta EGG [1].

Para que as regiões ortólogas múltiplas pudessem ser visualizadas, optamos por um algoritmo de alinhamento múltiplo de seqüências formadas por genes contíguos envolvidos nas regiões dois-a-dois encontradas.

O resultado da implementação da metodologia descrita, resultou em uma ferramenta web que se encontra encontra-se disponível no endereço *http://bagre.dct.ufms.br/~montera/mestrado*. O trabalho resultou na publicação [2].

Como futuras direções para o trabalho podemos citar:

1. Acréscimo de buscas mais refinadas - Algumas perguntas adicionais poderiam ser respondidas pelo BAGRE, como por exemplo:
    - Dado um gene g do genoma G, a quais outros genomas ele pertence?
    - Dado um conjunto de genes contíguos de G, em quais outros genomas esse conjunto pode ser encontrado?

2. Melhor visualização das ROMs - o alinhamento múltiplo estrela peca por acrescentar muitos buracos nas seqüências já inseridas. Deve haver maneiras de melhorar, ou minimizar o número de buracos acrescentados.

3. Algoritmo de cliques maximais - o algoritmo proposto neste trabalho não é eficiente. Apesar de não estarmos preocupado com a eficiência do algoritmo, uma vez que as instâncias em geral são factíveis, o estudo de heurísticas específicas para o problema, parece uma direção interessante.

4. Administração do sistema - A inclusão de um novo genoma no conjunto de genomas possíveis de comparação envolve a comparação deste com todos os outros genomas já incluídos no banco de dados. Como as operações de comparação dois-a-dois e inclusões no banco de dados não foram automatizadas, o usuário fica limitado a realizar comparações apenas entre os genomas disponibilizados pela ferramenta.

Referências Bibliográficas

[1] N.F. Almeida. *Ferramentas para comparação genômica*. Tese de Doutoramento, IC-UNICAMP, 2002.

[2] N.F. Almeida e L. Montera. Determinação de regiões ortólogas múltiplas. In S. Lifschitz, N.F. Almeida, G. Pappas, e R. Linden, editores, *Second Brazilian Workshop on Bioinformatics*, páginas 129–132. SBC, December 2003.

[3] N.F. Almeida, J. C. Setubal, e M. Tompa. On the use of don't care regions for protein sequence alignment. Relatório Técnico 99-07, Institute of Computing, University of Campinas, Brazil, 1999.

[4] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, e D.J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[5] T.H. Cormen, C.E. Leiserson, e R.L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, Cambridge,MA/New York, 1990.

[6] K. Cummings e W. Klug. *Concepts of Genetics*. Prentice Hall, New Jersey, USA, 5^{th} edição, 1997.

[7] A.C. Rasera da Silva, J.C. Setubal, e N.F. Almeida et al. Comparison of the genomes of two *xanthomonas* pathogens with differing host specificities. *Nature*, 417(6887):459–463, 2002.

[8] M. Dayhoff, R. Schwartz, e B. Orcutt. A model of evolutionary change in proteins. In M.O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, páginas 345–352. Natl. Biomed. Res. Found., 1978.

[9] M.V. de Souza, F.A. Torres, C.A. Ricart, W. Fontes, e M.A. Silva. *Gestão da vida?: genoma e pós-genoma*. Bluhm - UnB, 2001.

[10] A.L. Delcher, S. Kasif, R.D. Fleischmann, O. White J. Peterson, e S.L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.

[11] M. S. S. Felipe, M. E. M. T. Walter, M. M. Brígido, e N. F. Almeida Jr. et al. Transcriptome characterization of the dimorphic and pathogenic fungus paracoccidioides brasiliensis by est analysis. *Yeast*, 20:263–271, 2003.

[12] W. Fujibuchi, H. Ogata, H. Matsuda, e M. Kanehisa. Automatic detection of conserved gene clusters in multiple genomes by graph comparison and P-quasi grouping. *Nucleic Acids Research*, 28:4029–4036, 2000.

[13] P. Green. Genome sequence analysis. Lecture Notes - University of Washington, outubro 1996. (web site: http://www.genome.washington.edu/MBT599C/).

[14] D. Gusfield. Efficient methodos for multiple sequence alignment with guaranteed error bounds. *Bull. Math. Biol.*, 55:54–141, 1993.

[15] D. Gusfield. *Algorithms on Strings, Trees, and Sequences. Computer Science and Computational Biology*. Cambridge University Press, 1997.

[16] S. Henikoff e J. Henikoff. Amino acid substitution matrices from protein blocks. In *Proc. Natl. Acad. Sci. USA*, volume 89, páginas 10915–10919. 1992.

[17] M. Kanehisa e S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28:27–30, 2000.

[18] N. Kyrpides. Genomes online database (GOLD): a monitor of complete and ongoing genome projects world wide. *Bioinformatics*, 15:773–774, 1999.

[19] A. Lehninger. *Bioquímica*, volume 4. Edgard Blücher, 1977.

[20] B. Lewin. *Genes V*. Oxford University Press, Oxford, 1994.

[21] D. Liolios, A. Bernal, e N. Kyrpides. Genomes online database (gold). web site. Http://ergo.integratedgenomics.com/GOLD.

[22] S. Needleman e C. Wunsch. A general method applicable to the search for similarity in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[23] Nomenclature Comittee of the International Union of Biochemistry NC-IUB. Nomenclature for incompletely specified bases in nucleic acid sequences– recomendations 1984. *Eur. J. Biochem.*, 150:1–5, 1985.

[24] H. Ogata, W. Fujibuchi, S. Goto, e M. Kanehisa. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 28:4021–4028, 2000.

[25] M. C. F. Oliveira, M. A. S. Turine, e P. C. Masiero. A statechart model for hypertext. *CM Transactions on Information System*, July 2000.

[26] P.A. Pevzner. *Computational Molecular Biology*. MIT Press, 2000.

[27] S.L. Salzberg, D.B. Searls, e S. Kasif, editores. *Computational Methods in Molecular Biology*, volume 32 de *New comprehensive biochemistry*. Elsevier, Netherlands, 1998.

[28] R. Schwartz e M. Dayhoff. Matrices for detecting distant relationships. In M.O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, páginas 353–358. Natl. Biomed. Res. Found., 1978.

[29] J. C. Setubal e J. Meidanis. *Introduction to computational molecular biology*. PWS Publishing Co., 1997.

[30] J.C. Setubal e N.F. Almeida. Detection of related genes in procaryotes using syntenic regions. In *DIMACS Workshop on Whole Genome Comparison*. DIMACS Center, Rutgers University, February 2001.

[31] M. A. Van Sluys, J. C. Setubal, e N. F. Almeida Jr. et al. Comparative analyses of the complete genome sequences of pierce's disease and citrus variegated chlorosis strains of xylella fastidiosa. *Journal of Bacteriology*, 185:1018–1026, 2003.

[32] T. Smith e M. Waterman. Identification of common molecular subsequences. *Journal of Moleular Biology*, 147:195–197, 1981.

[33] B. Snel, G. Lehmann, P. Bork, e M.A. Huynen. STRING: a web-server to retrieve and display the repeatedly occuring neighbourhood of a gene. *Nucleic Acids Research*, 28, 2000.

[34] M. Suyama e P. Bork. Evolution of prokaryotic gene order: genome rearrangements in closely related species. *Trends in Genetics*, 17(1):10–13, January 2001.

[35] J. Tamames. Evolution of gene order conservation in prokaryotes. *Genome Biology*, 2(6), 2001.

[36] M.A.S. Turine. *HMBS: Um Modelo Baseado em Statecharts para Especificação Formal de Hyperdocumentos*. Tese de Doutoramento, Instituto de Física de São Carlos, USP, Junho 1998.

[37] M. Waterman. *Introduction to Computational molecular biology*. Chapman and Hall, 1995.

[38] D.W. Wood, J.C. Setubal, N.F. Almeida, e et al. Sequencing and analysis of the agrobacterium tumefaciens genome. In *10th Int'l congress on Molecular plant-microbe interactions*. 2001. Madison, WI (poster).

[39] D.W. Wood, J.C. Setubal, e N.F. Almeida et al. The genome of *agrobacterium tumefaciens*: insights into the evolution and evolution of a natural genetic enginner. *Science*, 294:2317–2323, December 2001.