BlastPhen

Agrupamento por similaridade com genomas completos

Ricardo Nishikido Pereira
ricnp at linux ime usp br

Orientadores:
Paolo Zanotto (ICB) e Marco Dimas Gubitoso (IME)

Universidade de São Paulo

São Paulo, 2004

Resumo

Esta monografia descreve uma iniciação científica na área de bioinformática, orientada pelos professores Paolo Zanotto[1] e Marco Dimas Gubitoso[2]. Além dos aspectos técnicos da iniciação científica, também está inclusa uma análise pessoal do projeto e do curso de Ciência da Computação.

A iniciação científica consistiu na elaboração de um programa que implementa técnicas de agrupamento por similaridade de genomas completos, utilizando medidas estatísticas e métodos de comparação de distribuições. A idéia principal consiste na utilização de genomas completos em oposição às técnicas tradicionais de reconstrução filogenética, que se baseam em genes.

[1]Instituto de Ciências Biomédicas
[2]Instituto de Matemática e Estatística

1 Introdução

A análise filogenética de uma família de ácidos nucleicos ou proteínas é a determinação de como essa família pode ter sido derivada durante a sua evolução. As relações evolucionárias entre as seqüências são mostradas colocando-se as mesmas como ramos externos de uma árvore. As relações entre os ramos internos refletem o grau de relacionamento entre diferentes seqüências. Por exemplo, duas seqüências muito parecidas serão colocadas como ramos externos vizinhos que estarão unidos por um ramo comum. O objetivo da análise filogenética é descobrir todas as relações entre os ramos de uma árvore e os comprimentos desses ramos.

Esta análise permite o estudo das mudanças ocorridas durante a evolução de diferentes organismos e a evolução de uma família de seqüências. Quando uma família de genes é encontrada em um organismo ou em um grupo deles, relações filogenéticas entre os genes podem ajudar a prever quais deles podem ter funções equivalentes. Essas previsões podem então ser testadas por experimentos genéticos. A análise filogenética também pode ser utilizada para rastrear mudanças ocorrendo em organismos de rápida evolução, como os vírus. O levantamento dos tipos de alterações ocorridas em uma população pode ser uma importante fonte de informação para a epidemiologia.

Técnicas estatísticas sofisticadas estão disponíveis para inferência filogenética, como o implementado no método de máxima verossimilhança (descrito em [7]). Contudo, esses métodos não consideram o genoma como um todo, mas apenas alguns genes. E isso pode impor um problema quando são feitas tentativas de integrar esses dados com os obtidos através de inferências baseadas em alinhamento de genes.

Alternativamente, podemos comparar genomas e construir distribuições a partir de medidas de similaridade entre eles. Essas distribuições são comparadas e diversas de suas características são estudadas com relação à sua utilidade para agrupamento de genomas durante a reconstrução filogenética.

Neste projeto propomos o *BlastPhen*, um programa que implementa técnicas de agrupamento por similaridade de genomas completos, utilizando medidas estatísticas e métodos de comparação de distribuições.

2 Objetivos

- Desenvolvimento de um programa (*BlastPhen*) que constrói uma matriz de similaridade entre diferentes organismos, através de comparações de distribuições estatísticas, permitindo seu uso para genomas complexos.

- Encontrar a métrica adequada para ser utilizada pelo programa.

- Utilizar o programa desenvolvido para formar árvores filogenéticas de conjuntos de vários organismos.

3 Materiais e métodos

O projeto foi desenvolvido no *LEMB* (Laboratório de Evolução Molecular e Bioinformática) do ICB-USP. O responsável pelo laboratório é o Prof. Dr. Paolo Zanotto.

O parque computacional disponível para a execução do programa é constituído por computadores IBM/PC, Macintosh, ALPHA/LINUX e Sun.

Os seguintes conjuntos de vírus (cujas filogenias são conhecidas) foram selecionados para comparar com os resultados do *BlastPhen*:

- **Adenoviridae**
  20 vírus
  tamanho médio: 33 762 pares de bases

- **Baculoviridae**
  26 vírus
  tamanho médio: 130 713 pares de bases

- **Herpesviridae**
  31 vírus
  tamanho médio: 156 236 pares de bases

- **Poxviridae**
  22 vírus
  tamanho médio: 190 509 pares de bases

O *BlastPhen* foi escrito na linguagem Perl, o que facilitou o tratamento de arquivos e a interação com outros programas. O pacote *BioPerl* (disponível em `http://bioperl.org`) foi utilizado para lidar com os arquivos contendo os genomas.

Os genomas são passados ao *BlastPhen* no formato *FASTA*. Uma seqüência no formato FASTA começa com uma linha de descrição, seguida por linhas que contém a seqüência propriamente dita.

A linha de descrição começa com o símbolo "¿". A palavra logo após este símbolo é considerada o "ID" (nome) da seqüência, e o resto da linha é a descrição. A seqüência acaba quando uma nova linha começando por "¿" é encontrada ou o arquivo acaba.

Exemplo de um arquivo FASTA com duas seqüências:

```
>Example1 envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
>Example2 synthetic peptide
HITREPLKHIPKERYRGTNDTLSPQIESIWAAELDRYKLVKTNCSNVS
```

3.1 Medidas estatísticas

As primeiras medidas estudadas foram média, mediana e moda. Foram também estudadas as seguintes medidas de comparação de distribuições:

- Kullback-Leibler divergence

$$\mathcal{D}(p_1 \parallel p_0) = \int p_1(x) log \frac{p_1(x)}{p_0(x)} dx$$

  Como a distância de Kullback-Leibler não é simétrica, foi utilizada a média harmônica para simetrizá-la (conforme sugerido em[5]):

$$\frac{1}{\mathcal{R}(p_0, p_1)} \equiv \frac{1}{\mathcal{D}(p_1 \parallel p_0)} + \frac{1}{\mathcal{D}(p_0 \parallel p_1)}$$

- Skew divergence [6]

$$\mathcal{S}_\alpha(q, r) = \mathcal{D}(r \parallel \alpha q + (1 - \alpha) r)$$

  Onde $\mathcal{D}$ é a distância de Kullback-Leibler e α é um fator de correção.

  Como a Skew não é simétrica, utilizamos o mesmo método de simetrização da Kullback-Leibler.

- Chernoff

$$\mathcal{C}(p_0, p_1) = \max_{0 \le t \le 1} -log\mu(t), \qquad \mu(t) = \int [p_0(x)]^{1-t} [p_1(x)]^t dx$$

- Bhattacharyya

$$\mathcal{B}(p_0, p_1) = -\log \mu(\tfrac{1}{2})$$

  Onde μ é a função definida na distância de Chernoff.

Notemos que as métricas citadas acima referem-se a distribuições contínuas, enquanto que neste projeto lidamos com distribuições discretas (de *scores* do *Blast*). Portanto, são criados histogramas para agrupar os *scores* e as integrais foram substituídas por somatórios. Um problema que pode ocorrer durante a construção dos histogramas é que, se os genomas em questão forem muito diferentes, os *scores* de sua comparação serão muito baixos. Assim, ao compararmos essa distribuição, que anteriormente chamamos de D_2, com a distribuição D_1 poderá acontecer de a primeira resumir-se a apenas uma classe do histograma. Para evitar esse problema, construímos o histograma de forma que a distribuição de menor *footprint* seja distribuída em 10 classes.

Um problema da distância de Kullback-Leibler é que ela não aceita que a função p_0 assuma valores nulos. Como não lidamos com distribuições contínuas mas sim com histogramas, eventualmente classes do histograma da distribuição p_0 assumem valores nulos. Por isso, tivemos que adaptar os histogramas para que nenhum valor fosse nulo, mas assumisse um valor baixo. Esse valor é calculado como sendo 5% do valor mínimo que uma classe possui, considerando as duas distribuições sendo comparadas.

É exatamente esse tipo de problema que a Skew procura resolver. Utilizando o parâmetro α, notamos que o problema é resolvido, sem termos que alterar os histogramas. A Skew defende a idéia de que é melhor utilizarmos uma versão aproximada da Kullback-Leibler do que utilizar a Kullback-Leibler em distribuições adaptadas.

Podemos observar as semelhanças entre as distâncias de Chernoff e Bhattacharyya. De fato, a distância de Bhattacharyya é um caso particular da de Chernoff, com o parâmetro t igual a $\frac{1}{2}$. Essas distâncias foram testadas quanto à proximidade de seus resultados e foi verificado, empioricamente, que realmente a Bhattacharyya é uma boa aproximação da Chernoff. Um problema da Chernoff é que ela é computacionalmente difícil de ser calculada, enquanto que a Bhattacharyya é muito simples.

Embora as distâncias de Chernoff e Bhattacharyya aceitem que as distribuições assumam o valor zero em alguns pontos, optamos por fazer a mesma adaptação nos histogramas feita para a Kullback-Leibler. Isso porque sempre que uma distribuição for zero em um ponto, o elemento correspondente no somatório será zero, e como isso pode acontecer muitas vezes perderíamos muita informação.

3.2 *Blast*

O *Blast* (Basic Local Alignment Search Tools) [2], desenvolvido pelo *NCBI* (National Center for Biotechnology Information), é um conjunto de algoritmos de comparação de seqüências de nucleotídeos e amino-ácidos. Segmentos

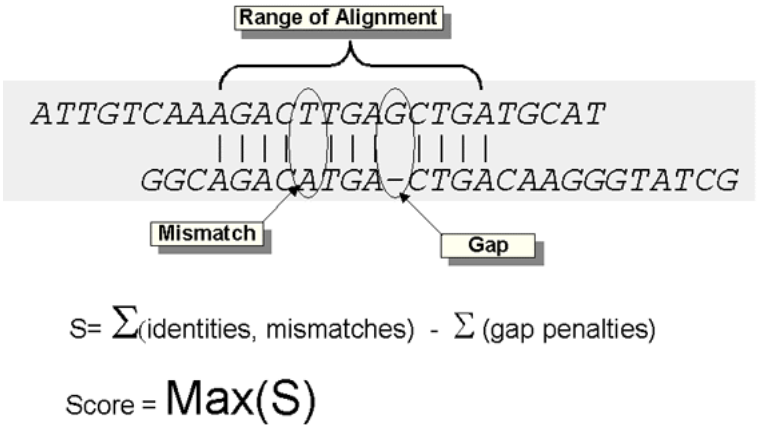

Figura 1: Cálculo do *raw score*

de uma seqüência são comparados com segmentos de outras seqüências e são atribuídos pontos (*scores*) que indicam o grau de similaridade entre os segmentos. Quanto mais alta é a pontuação, maior é o grau de similaridade. Assim, ao final de uma execução do *Blast* temos, para cada par de seqüências, zero ou mais *scores*, dependendo do quão semelhantes elas são.

A seguir serão descritos alguns termos relevantes ao estudo do *Blast*:

- Identidade (*identity*): segmento no qual duas seqüências são invariantes. Para nucleotídeos, as identidades valem +5, enquanto que para amino-ácidos seus valores são dados por uma matriz de substituição.

- Substituição (*substitution*): presença de bases diferentes em uma posição de um alinhamento. Para nucleotídeos, uma substituição vale -4, para amino-ácidos o valor depende da matriz de substituição utilizada.

- *Gap*: espaço produzido em um alinhamento para compensar inserções e remoções em uma seqüência em relação à outra. Um *score* de $-a$ é cobrado pela existência de um gap e um *score* de $-b$ é cobrado por cada resíduo do gap. Assim, um gap de k resíduos vale

$$-(a + kb)$$

  Os valores desses custos são calculados empiricamente; em geral atribui-se um valor alto para a (10–15) e um baixo para b (1–2).

- *Raw score* (figura 1): o *score* de um alinhamento, calculado como a soma de *scores* de substituição e de gaps.

$$S = \sum(\text{identidades}, \text{substituicoes}) - \sum(\text{custo dos gaps})$$

- *Bit score*: derivado do raw *score*; leva em conta as propriedades estatísticas do sistema de pontuação (matriz de substituição e custo dos gaps).

$$S' = \frac{\lambda S - lnK}{ln2}$$

  Onde λ e K dependem do sistema de pontuação (matriz de substituição e tamanho das seqüências).

  Por ser uma versão normalizada do *raw score*, o *bit score* pode ser utilizado em comparações com *scores* de outros alinhamentos.

4 Desenvolvimento

O programa passa as seqüências ao *Blast* para comparar os genomas uns com os outros, de maneira par a par. O *Blast* compara segmentos de um genoma com segmentos de outro, e a cada uma dessas comparações é atribuído um valor (*score*) que indica a similaridade desses segmentos. Quanto mais alto o *score*, maior a similaridade. O *BlastPhen* utiliza os resultados do *Blast* para construir distribuições de *scores* por sua freqüência para cada par de genomas. (Para mais informações sobre o *Blast* veja a seção 3.2 ou [2].)

As distribuições são analisadas através das medidas estatísticas descritas em 3.1.

Para as medidas média, mediana e moda, submetemos dois genomas ao *Blast*, obtendo uma distribuição de *scores* por sua freqüência. Calculamos então a média, a mediana e a moda dessa distribuição, que são candidatas a distância entre os genomas.

Para as métricas de comparação de distribuições, o procedimento de cálculo da estimativa de distância genética é um pouco mais complexo. Seja G_1 um genoma. Para calcular as distâncias entre G_1 e os demais genomas procederemos da seguinte maneira. Primeiramente comparamos G_1 com ele mesmo, através do *Blast*, obtendo então uma distribuição de *scores* D_{11}. Em seguida, para descobrir a distância entre G_1 e G_n compararemos os genomas com o *Blast*, obtendo outra distribuição de *scores* D_{1n}, e então compararemos as duas distribuições obtidas (D_{11} e D_{1n}).

Após compararmos todos os genomas e utilizarmos as métricas estatísticas mencionadas, obtemos uma matriz de distâncias (para cada métrica). Observemos que essa matriz não é simétrica (isso se deve à maneira como o *Blast* calcula os *scores*). Para simetrizar a matriz, fazemos a média aritmética dos elementos correspondentes. Depois, subtraímos o elemento da diagonal principal dos demais elementos da linha. Com isso obtemos uma matriz simétrica que, além disso, terá zeros em sua diagonal principal. A explicação para esse procedimento é que podemos entender os valores da diagonal como sendo a menor distância que um genoma pode ter até outro, ou seja, a menor distância que um genoma G pode ter a um outro qualquer é a distância que ele tem até si próprio. Pela maneira como são calculados os *scores* e pelo método como o *BlastPhen* calcula as distâncias, um genoma não terá distância zero até si próprio, ao contrário do que esperaríamos. Encaramos esse valor como um fator de ajuste, por isso o subtraímos das demais distâncias entre esse genoma e os outros. Através de testes com as bases de dados mencionados observamos que esse procedimento é válido.

Opcionalmente, o *BlastPhen* permite a criação de *bonsais*. Os bonsais são equivalentes a cliques da teoria dos grafos: são grupos de vértices dois a dois adjacentes. Na genética, um bonsai é um grupo de organismos que possui uma relação ancestral (i.e. distância finita) com todos os outros organismos desse grupo. Essa separação de organismos é possível porque às vezes o *Blast* não pode detectar *scores* significativos entre dois ou mais organismos, pois seus genomas podem ser muito diferentes.

O algoritmo para criação de bonsais é simples. Tendo um conjunto de organismos em mãos, que inicialmente é o conjunto de todos os organismos passados ao *BlastPhen*, procuramos por uma distância infinita entre dois elementos. Se acharmos uma distância infinita entre O_1 e O_2, separamos os organismos em dois grupos: um deles contendo todos os organismos menos O_1 e o outro contendo todos menos O_2.

O problema é que esse algoritmo é exponencial. Assim, se existirem muitos bonsais essa etapa pode ser muito demorada.

Depois de separar os organismos em Bonsais, o *BlastPhen* gera diversas matrizes de distâncias, uma para cada bonsai.

Tendo a(s) matriz(es) de distâncias do *BlastPhen*, o usuário pode utilizar o programa *neighbor* (do pacote de programas *Phylip*, disponível em [3]), ou qualquer outro programa similar, para gerar as árvores filogenéticas. Neste projeto as árvores foram geradas com o programa citado utilizando a técnica de UPGMA (mais informações sobre essa técnica em [5]).

Com o intuito de otimizar o desempenho do *BlastPhen*, seu código foi programado para utilizar diversas máquinas com diferentes números de processadores. A idéia inicial era utilizar o padrão *MPI* [1] para realizar a

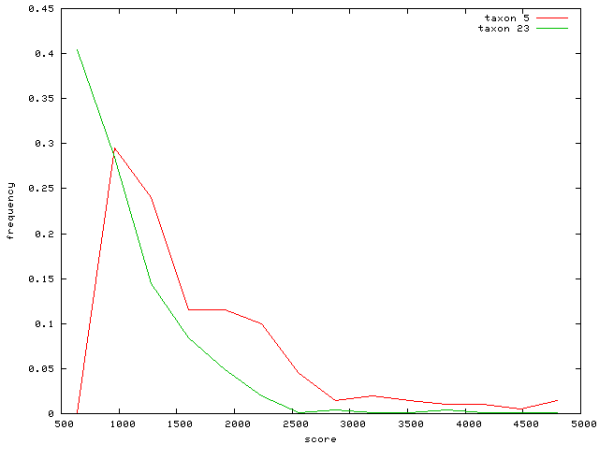

Figura 2: Distribuição de *scores* de organismos parecidos geneticamente

comunicação entre as máquinas. Tentamos utilizar a implementação *mpich* [4], mas ela não se comportou bem com algumas arquiteturas. Tentamos então o *lam-mpi* [8], mas ela não conseguia lidar com firewalls, algo imprescindível no projeto, pois uma das máquinas se localizava no IME enquanto que as outras se encontravam no ICB. Por fim, optou-se por utilizar *sockets* para realizar a comunicação.

A parte do processo que se beneficia da paralelização é a submissão dos genomas ao *Blast*. Um processo servidor controla a distribuição de genomas aos processos clientes, que por sua vez os submetem ao *Blast*. Quando todas as seqüências forem utilizadas, o processo servidor reúne os resultados e o *BlastPhen* prossegue com seus cálculos.

5 Resultados

Das técnicas estatísticas estudadas a que mostrou melhores resultados foi o cálculo das medianas das distribuições. Os métodos de comparação de distribuições analisados não se comportaram bem devido a falta de resolução das curvas, que dificultou as comparações.

A figura 2 mostra uma situação ideal, onde temos dados suficientes para comparar as distribuições. Já na figura 3, podemos ver que a distribuição de *scores* resultante da comparação de dois organismos muito diferentes geneticamente é muito pobre. Assim, a comparação com distribuições desse tipo tende a ser muito pouco informativa.

Os dados obtidos com o *BlastPhen* foram analisados e comparados com os resultados conseguidos com técnicas tradicionais de análise filogenética e

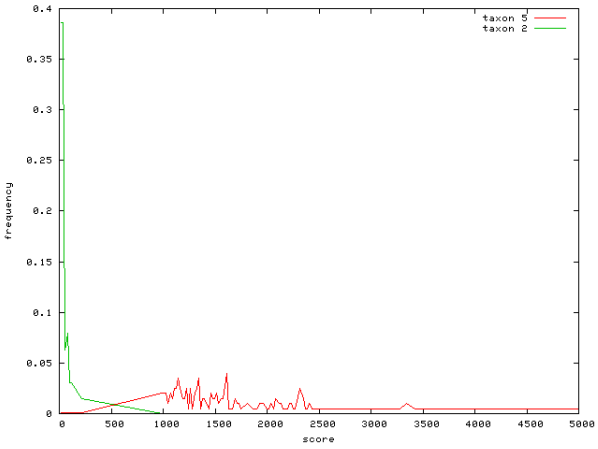

Figura 3: Distribuição de *scores* de organismos muito diferentes geneticamente

a conclusão foi que o método empregado pelo *BlastPhen* é correto e eficiente. Podemos verificar tal fato observando os gráficos que comparam os resultados do *BlastPhen* com filogenias já conhecidas, nas figuras 4, 5, 6, 7, 8, 9, 10, 11.

6 Experiência pessoal

6.1 Desafios

Por situar-se em uma área multidisciplinar e principalmente por trabalhar diretamente com pessoas de um laboratório de biologia, esta iniciação científica trouxe diversos desafios para mim.

Primeiramente, tive que lidar constantemente com pessoas que não eram da minha área, o que me ensinou muitas coisas. A maneira diferente de pensar e trabalhar abriu minha mente para além do mundo da computação. Tive que me habituar ao cotidiano do mundo científico e aprender a me comunicar com biólogos. Uma diferença que percebi entre os cientistas da computação e os biólogos é que enquanto que os primeiros estão muitas vezes preocupados com o formalismo de um algoritmo, os últimos estão muito mais preocupados com algo mais imediato, como um programa que mostre resultados.

Tive também que aprender alguns conceitos de biologia molecular, o que foi um pouco difícil pois não tinha muita base nessa área, somente o pouco da biologia do ensino médio de que me recordava. Mais uma vez, foi bom aprender coisas completamente novas para mim; uma das vantagens de se estudar em uma boa universidade é a possibilidade de interagir com áreas diferentes, o que permite uma formação sem igual.

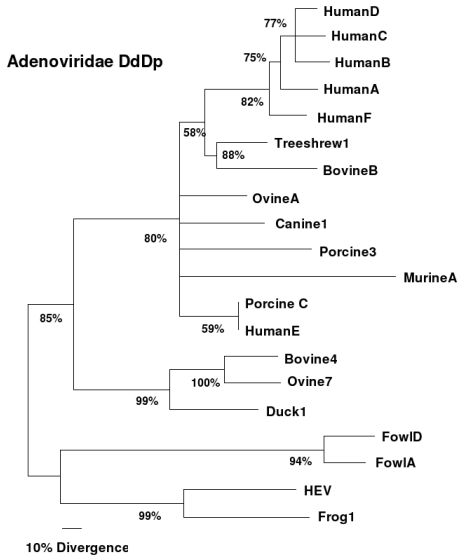

Figura 4: Adeno vírus - árvore construída com técnicas tradicionais de reconstrução filogenética

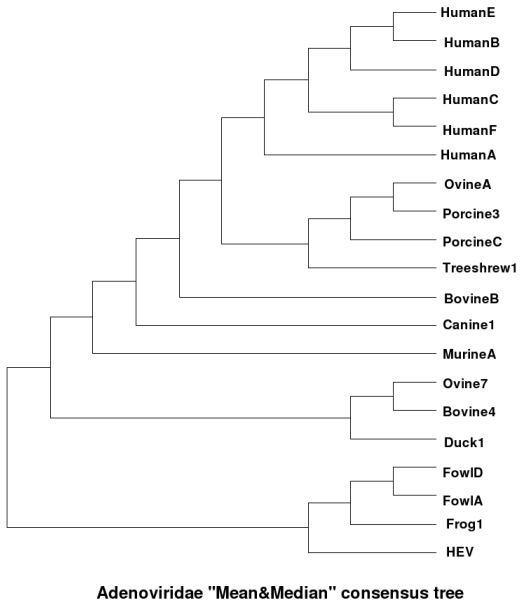

Figura 5: Adeno vírus - árvore construída com dados do *BlastPhen*

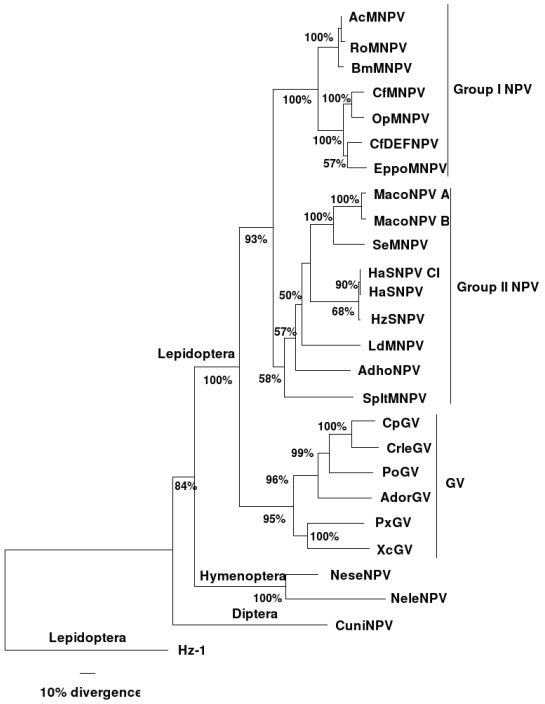

Figura 6: Báculo vírus - árvore construída com técnicas tradicionais de reconstrução filogenética

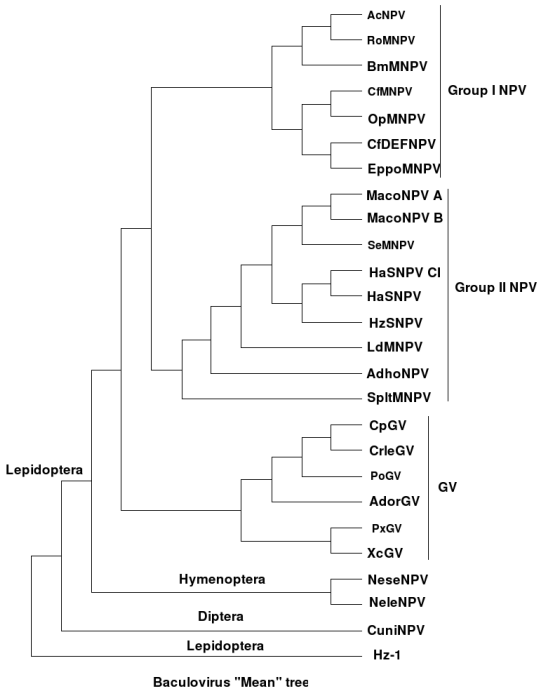

Figura 7: Báculo vírus - árvore construída com dados do *BlastPhen*

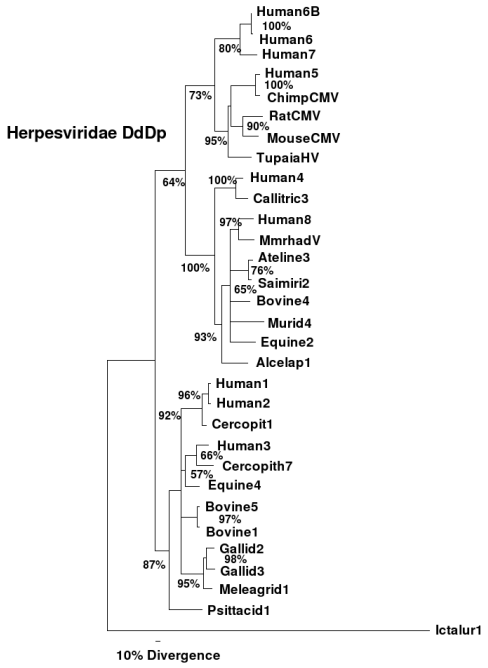

Figura 8: Herpes vírus - árvore construída com técnicas tradicionais de reconstrução filogenética

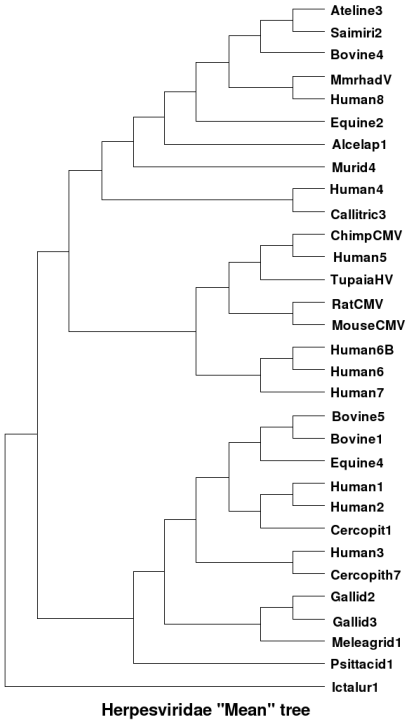

Figura 9: Herpes vírus - árvore construída com dados do *BlastPhen*

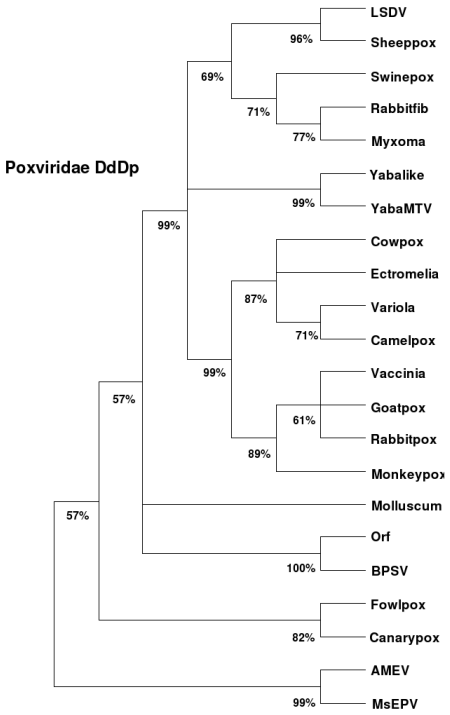

Figura 10: Pox vírus - árvore construída com técnicas tradicionais de reconstrução filogenética

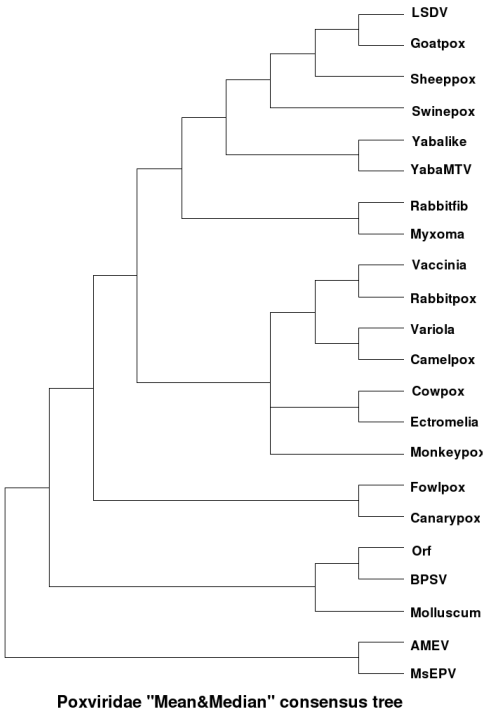

Figura 11: Pox vírus - árvore construída com dados do *BlastPhen*

Apesar disso, existiram momentos frustrantes durante a execução do projeto. Algumas fases, por exemplo, consistiam de extensivos testes e ajustes de parâmetros, o que foi entediante e exaustivo. A falta de resultados nessa fase e a falta de certeza de sucesso dos testes eram desanimadores. Mais decepcionante ainda foram os estudos que se mostraram inúteis ao projeto, como algumas das métricas pesquisadas. Na fase de paralelização do programa, tive que testar diversas ferramentas pois muitas mostraram problemas para serem utilizadas nos equipamentos disponíveis ou no tipo de rede que tínhamos.

6.2 BCC e a iniciação científica

A seguir listarei as disciplinas do BCC mais relevantes para a realização da minha iniciação científica:

- Laboratório de Programação II (MAC 0242): Além de aprender a linguagem PERL, esta disciplina me ensinou a lidar com projetos relativamente extensos.
- Introdução à Probabilidade e à Estatística I (MAE 0121) e II (MAE 0212):

Foram muito importantes pois tive que lidar com diversos conceitos estatísticos durante o projeto.

- Cálculo Diferencial e Integral I (MAT 0111): Uma base em cálculo foi necessária para o entendimento de alguns conceitos matemáticos, como por exemplo integrais.

- Algoritmos em Grafos (MAC 0328): Utilizei conceitos de grafos para lidar com cliques. Apesar desse assunto não ter sido tratado em aula, o que aprendi foi uma base para buscar os conhecimentos de que precisava, bem como para a implementação do algoritmo que lida com cliques.

- Princípios de Desenvolvimento de Algoritmos (MAC 0122) e Análise de Algoritmos (MAC 0338): Muito úteis para desenvolver e compreender melhor os algoritmos do programa.

Acredito que também teria sido proveitoso cursar a disciplina Biologia Computacional e alguma disciplina básica de biologia.

Ao aplicar os conceitos vistos no BCC na prática senti-me gratificado, vendo os resultados surgirem e serem utilizados. É diferente de fazer um trabalho como um EP, pois neste caso às vezes não nos sentimos muito motivados já que o programa não produzirá resultados práticos. De qualquer forma, senti que deveria ter me empenhado mais em algumas disciplinas.

6.3 As relações de trabalho

Uma das diferenças que eu notei entre as relações de trabalho com outros alunos do BCC e com as pessoas do laboratório foi que no BCC é mais fácil se comunicar, pois todos falam "o mesmo idioma". Já com as pessoas do laboratório, tinha que me policiar para não ser muito técnico e também certificar-me de que realmente entendia o que me era dito. Essa dificuldade de comunicação é dupla, afinal eles não estavam habituados com alguns dos meus termos nem eu estava acostumado com os termos deles. Mas com o tempo atinge-se um equilíbrio.

O fato de ter tido dois orientadores nessa iniciação científica foi uma experiência engrandecedora. Além de ver como pessoas de áreas diferentes se relacionam, pude aprender como trabalham e pensam.

Referências

[1] Message passing interface forum web site http://www.mpi-forum.org/.

[2] Ncbi blast web site http://www.ncbi.nlm.nih.gov/blast/.

[3] Phylip home page http://evolution.genetics.washington.edu/phylip.html.

[4] William Gropp and Ewing Lusk. Instalation and user's guide to mpich, a portable implementation of mpi — version 1.2.6 — the ch_p4 device for workstation networks. Technical report, University of Chicago, http://www-unix.mcs.anl.gov/mpi/mpich/.

[5] Don H. Johnson and Sinan Sinanović. Symmetrizing the kullback-leibler distance. Technical report, Rice University, Houston, TX, march 2001.

[6] Lilian Lee. On the effectiveness of the skew divergence for statistical language analysis. Technical report, Cornell University, Ithaca, New York, 2001.

[7] David W. Mount. *Bioinformatics: sequence and genome analysis.* Cold Spring Harbor Laboratory Press, 2001.

[8] The Lam/MPI Team. Lam/mpi user's guide — version 7.1.1. Technical report, Open Systems Lab, http://www.lam-mpi.org, 2004.