PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 17/07

Avaliação do E-value para Execução do BLAST sobre Bases de Dados Fragmentadas

Daniel Xavier de Sousa
Sérgio Lifschitz

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 17/07 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena Julho, 2007

Avaliação do E-value para Execução do BLAST sobre Bases de Dados Fragmentadas *

Daniel Xavier de Sousa e Sérgio Lifschitz

dsousa@inf.puc-rio.br, sergio@inf.puc-rio.br

Abstract. The BLAST tool is widely used in bioinformatics research projects, helping with the biological sequences comparisons and homology search. Stand-alone executions, involving a single input sequence (query) and a relatively small database, no performance problems arise. However, with large databases, there is a need for execution approaches that achieve better execution times. An alternative would be parallel computing environments, and in most cases a distributed database allocation through the available machines. Particularly, when the sequence database is fragmented, correctness must be guaranteed in order to get output results that are consistent with the equivalent serial execution. This fundamental problem occurs due to the statistics used by BLAST in its heuristics, such as the e-value. Computations are directly related to the size of the database. When the latter is fragmented, the results obtained may not infer the same semantics as when BLAST is run in the conventional (serial) way. This paper aims at discussing the BLAST parallel evaluation on fragmented databases. Our goal is to support developers and users with respect to their decisions regarding configuration and parameters for BLAST execution in these situations.

Keywords: BLAST, Distributed Database, Similarity Statistics, E-value.

Resumo. A ferramenta BLAST é bastante utilizada em projetos de pesquisa na área de bioinformática, auxiliando no processo de comparação de seqüências biológicas e busca por homologias. Isoladamente, para apenas uma seqüência (consulta) de entrada e um banco de dados relativamente pequeno, não há problemas de desempenho. Porém, com bancos de dados mais volumosos, são necessárias abordagens de melhoria no tempo de execução. Para isto, pode-se optar por ambientes de programação paralela, quase sempre com a conseqüente distribuição dos dados pela máquinas disponíveis. No caso particular de bases de seqüências fragmentadas, é preciso garantir a correção da execução de forma que a avaliação em paralelo retorne resultados corretos, equivalentes semanticamente aos resultados de execução serial. Este problema fundamental ocorre devido às estatísticas utilizadas pelo BLAST em suas heurísticas, como o *e-value*. Os cálculos são realizados em função do tamanho da base de dados completa e quando esta é fragmentada, os resultados obtidos podem não permitir as mesmas conclusões obtidas pela execução serial convencional. Este artigo tem por objetivo discutir a avaliação paralela do BLAST sobre bases fragmentadas e apoiar desenvolvedores e usuários nas decisões que precisam ser tomadas para configuração e escolha de parâmetros de execução.

Palavras-chave: BLAST, banco de dados distribuídos, Estatística, E-value.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

1 Introdução

Uma atividade extremamente importante na biologia molecular é o alinhamento de seqüências biológicas [4]. O objetivo é realizar buscas por similaridades, que quando encontradas, indicam uma alta probabilidade de se encontrar funções análogas entres as seqüências.

Com o grande volume de dados oriundos dos projetos de sequenciamento, ferramentas de alinhamento se tornaram muito úteis. As ferramentas hoje disponíveis utilizam duas formas de alinhamento[4]: global e local. Na primeira, seqüências são totalmente comparadas, isto é, verificando se há similaridade entre seqüências inteiras. No alinhamento local, há uma busca por trechos de seqüência que sejam parecidos.

A ferramenta BLAST (Basic Local Alignment Search Tool)[11] utiliza heurísticas e algoritmos de programação dinâmica para obter os melhores alinhamentos locais, com um tempo de execução bem reduzido em relação aos programas até então conhecidos[8]. Mesmo assim, com o atual crescimento das bases de dados o BLAST tem se tornado demasiadamente lento.

Uma alternativa para acelerar o processo de execução da ferramenta BLAST é a utilização de aglomerados de computadores (cluster) acessando a base de dados de forma distribuída [1, 2]. Neste processo cada máquina de trabalho recebe um fragmento do banco e o compara com todas as seqüências de consulta.

Um motivo forte que justifica a boa aceitação da ferramenta BLAST pela comunidade científica, diz respeito à confiança dada para a semelhança identificada entre duas seqüências. Este grau de confiabilidade é o objeto de discussão deste trabalho, principalmente quando utiliza-se bancos de dados distribuídos não replicados.

No caso particular de bases de seqüências fragmentadas, é preciso garantir a correção da execução de forma que a avaliação em paralelo retorne resultados corretos, equivalentes semanticamente aos resultados de execução serial. Este problema fundamental ocorre devido às estatísticas utilizadas pelo BLAST em suas heurísticas, como o *e-value*. Neste sentido, foi realizado um estudo das alterações na pontuação dada pela estatística de alinhamento quando da alocação de bancos fragmentados e é sugerido um procedimento para obter valores estatísticos mais próximo dos originais, isto é, equivalentes à execuções seriais com toda a base de dados. Outros trabalhos mais específicos explicam como os valores estatísticos de alinhamento são utilizados pelo BLAST [4, 12, 13,].

O restante deste trabalho está organizado da seguinte forma: a Seção 2 resume como são definidas as estatísticas de alinhamento no BLAST, a Seção 3 mostra as alterações nas estatísticas considerando bases fragmentas e a Seção 4 explica o uso de parâmetros na correção destas estatísticas. Por fim, a Seção 5 conclui este trabalho e aponta direções para trabalhos futuros.

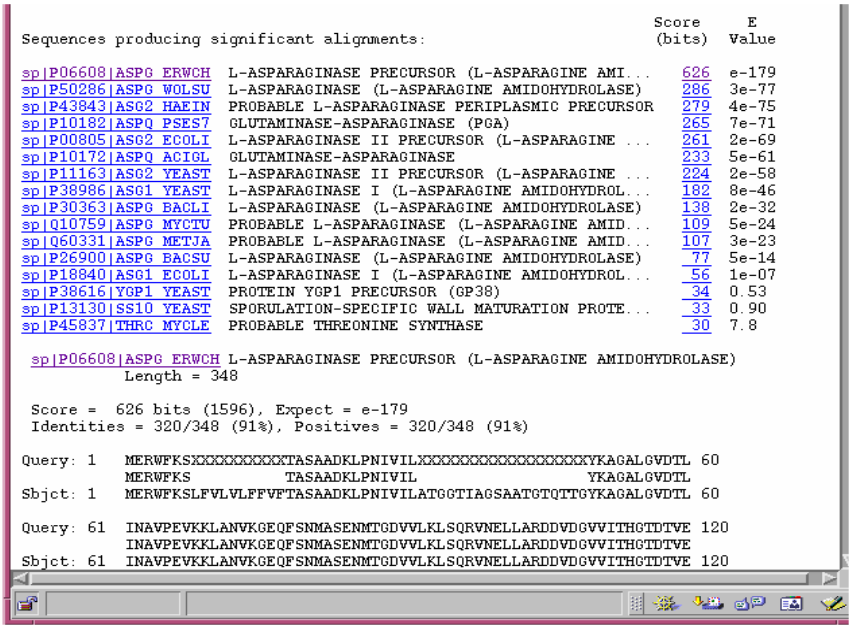
2 Estatísticas de Alinhamento

Para se utilizar a ferramenta BLAST, além dos vários parâmetros de entrada que podem regular a comparação, são solicitadas: a seqüências de consulta e a base de seqüências escolhida. A seqüência de consulta é comparada com todas as seqüências do

banco de dados e o BLAST retorna um relatório informando detalhes de similaridade das seqüências do banco. A Figura 1 ilustra um exemplo deste relatório de saída.

Antes de prosseguir, cabe observar que grande parte das informações contidas nesta Seção foram retiradas de trabalhos que tratam as estatísticas de alinhamento da ferramenta BLAST [4, 12].

Como pode ser observado na Figura 1, a primeira parte do relatório lista todas as seqüências da base de dados que possuem alta similaridade, também chamadas de *hits*, com a seqüência de consulta. Para cada seqüência da base de dados listada, ou cada *hit*, é associada uma pontuação (*score*) que quantifica o grau de similaridade entre as duas seqüências. Esta pontuação, assim como todas as seqüências similares, são obtidas a partir de heurísticas e aproximações que a ferramenta BLAST utiliza. Para dar confiabilidade aos seus resultados, ao lado de cada valor de *score* é fornecido o *e-value*. Para explicar o *e-value*, é necessário lembrar brevemente algumas características do algoritmo BLAST.



```
                                                                  Score     E
Sequences producing significant alignments:                       (bits)  Value

sp|P06608|ASPG_ERWCH  L-ASPARAGINASE PRECURSOR (L-ASPARAGINE AMI...   626  e-179
sp|P50286|ASPG_WOLSU  L-ASPARAGINASE (L-ASPARAGINE AMIDOHYDROLASE)    286  3e-77
sp|P43843|ASG2_HAEIN  PROBABLE L-ASPARAGINASE PERIPLASMIC PRECURSOR   279  4e-75
sp|P10182|ASPQ_PSES7  GLUTAMINASE-ASPARAGINASE (PGA)                 265  7e-71
sp|P00805|ASG2_ECOLI  L-ASPARAGINASE II PRECURSOR (L-ASPARAGINE ...   261  2e-69
sp|P10172|ASPQ_ACIGL  GLUTAMINASE-ASPARAGINASE                       233  5e-61
sp|P11163|ASG2_YEAST  L-ASPARAGINASE II PRECURSOR (L-ASPARAGINE ...   224  2e-58
sp|P38986|ASG1_YEAST  L-ASPARAGINASE I (L-ASPARAGINE AMIDOHYDROL...   182  8e-46
sp|P30363|ASPG_BACLI  L-ASPARAGINASE (L-ASPARAGINE AMIDOHYDROLASE)    138  2e-32
sp|Q10759|ASPG_MYCTU  PROBABLE L-ASPARAGINASE (L-ASPARAGINE AMID...   109  5e-24
sp|Q60331|ASPG_METJA  PROBABLE L-ASPARAGINASE (L-ASPARAGINE AMID...   107  3e-23
sp|P26900|ASPG_BACSU  L-ASPARAGINASE (L-ASPARAGINE AMIDOHYDROLASE)     77  5e-14
sp|P18840|ASG1_ECOLI  L-ASPARAGINASE I (L-ASPARAGINE AMIDOHYDROL...    56  1e-07
sp|P38616|YGP1_YEAST  PROTEIN YGP1 PRECURSOR (GP38)                   34  0.53
sp|P13130|SSI0_YEAST  SPORULATION-SPECIFIC WALL MATURATION PROTE...   33  0.90
sp|P45837|THRC_MYCLE  PROBABLE THREONINE SYNTHASE                     30  7.8

 sp|P06608|ASPG_ERWCH L-ASPARAGINASE PRECURSOR (L-ASPARAGINE AMIDOHYDROLASE)
          Length = 348

 Score =  626 bits (1596), Expect = e-179
 Identities = 320/348 (91%), Positives = 320/348 (91%)

Query: 1   MERWFKSXXXXXXXXXXTASAADKLPNIVILXXXXXXXXXXXXXXXXXXYKAGALGVDTL 60
            MERWFKS          TASAADKLPNIVIL                  YKAGALGVDTL
Sbjct: 1   MERWFKSLFVLVLFFVFTASAADKLPNIVILATGGTIAGSAATGTQTTGYKAGALGVDTL 60

Query: 61  INAVPEVKKLANVKGEQFSNMASENMTGDVVLKLSQRVNELLARDDVDGVVITHGTDTVE 120
            INAVPEVKKLANVKGEQFSNMASENMTGDVVLKLSQRVNELLARDDVDGVVITHGTDTVE
Sbjct: 61  INAVPEVKKLANVKGEQFSNMASENMTGDVVLKLSQRVNELLARDDVDGVVITHGTDTVE 120
```

Figura 1 – Exemplo de relatório fornecido pelo BLAST.

A ferramenta BLAST procura identificar palavras (subconjunto de caracteres pertencentes a uma seqüência) que são completamente semelhantes entre a seqüência de consulta e as seqüências do banco de dados. Estas palavras são estendidas em ambas às direções visando ampliar a região de similaridade. A todo alinhamento entre um par de palavras é dada uma pontuação que, todas somadas, definem um valor de *score* para o trecho considerado similar. A extensão das palavras ocorre até que o somatório das pontuações permaneça acima de um valor T dado. Uma região de alta similaridade com pontuação acima do valor T é chamada de HSP (do inglês, *High-scoring Segment Pair*).

Contudo, mesmo considerando um alto *score* entre duas seqüências, nem sempre podemos inferir um alto grau de homologia[1] entre elas. Devido à aleatoriedade no processo de alinhamento, podem ser seqüências que não tenham relacionamento algum. Dada a variabilidade entre as seqüências comparadas e as heurísticas do BLAST, utiliza-se assim o *e-value*, que procura fornecer aos usuários a segurança de que a pontuação dada para um determinado *hit* não ocorreu aleatoriamente.

O valor *e-value* de um *hit* para um dado *score* corresponde à probabilidade de se obter, com outra seqüência aleatória de mesmo tamanho e composição de letras, outro alinhamento com *score* igual ou superior. Desta forma, quanto mais próximo de zero for o *e-value*, mais confiável será a consulta. Por exemplo, para um *hit* com e-value=1, significa que para a mesma seqüência de consulta, comparada com outra seqüência aleatória de mesmo tamanho e composição, vai gerar um alinhamento com *score* igual ou maior ao obtido.

Um e-value para um *score* S é dado pela seguinte função [4]:

$$E(S) = mnKe^{-\lambda S} \qquad \text{[Equação 1]}$$

Onde *m* é o tamanho da seqüência de consulta e *n* é o tamanho das seqüências do banco de dados. Assim, *n x m* é o espaço de busca. *K* é uma constante obtida a partir das séries geométricas dependentes da probabilidade de ocorrência dos caracteres do *hit* e da pontuação entre as combinações das seqüências, ou seja, determinada pela composição do espaço de busca. Já o valor λ é uma constante referente às matrizes de substituição utilizadas para obter a pontuação entre os caracteres das seqüências.

A utilização pura do *score*, isto é, *score* não normalizado, pode gerar valores errados, pois os fatores de escala são arbitrários e não levam em consideração o sistema de pontuação e os parâmetros estatísticos[2]. Um *score* normalizado (*bit score*) S' é obtido a partir da fórmula:

$$S' = (\lambda S - \ln K)/\ln 2 \qquad \text{[Equação 2]}$$

Com o *bit score* é possível obter o *e-value* da seguinte forma:

$$E(S) = mn2^{-S'} \qquad \text{[Equação 3]}$$

Assim, de posse do S', é possível obter o *e-value* somente utilizando o espaço de busca de *m* e *n*.

Entretanto, na ferramenta BLAST, o espaço de busca não corresponde exatamente ao tamanho da seqüência de consulta multiplicada pelo tamanho de todas as seqüências da base de dados. No relatório de saída do BLAST podemos perceber no rodapé

---

1 Por homologia entende-se semelhança entre estruturas de diferentes organismos, devida unicamente a uma mesma origem embriológica. As estruturas homólogas podem exercer ou não a mesma função.

2 Utilizar *score* não normalizado é o mesmo que informar o peso de algum objeto sem dizer a unidade de massa, como: kg ou mg.

(exemplo na Figura 2) os valores para os tamanhos reais da seqüência e os tamanhos efetivos. Valores reais são os valores exatos, seja da seqüências de consulta ou da base de dados. Os efetivos são os valores alterados a partir dos valores reais, que serão utilizados pela ferramenta BLAST. No caso do *e-value*, a partir da Equação 1, os parâmetros utilizados são os tamanhos efetivos.

No processo de extensão para encontrar as regiões de HSP, o BLAST não realiza extensões das palavras que se situam próximas do fim das seqüências. Pois os HSP devem ter um tamanho mínimo. A ferramenta BLAST calcula o tamanho destas extremidades que não fazem parte do espaço de busca, e que não produzirão alinhamentos significativos. O tamanho mínimo *I* para estas regiões, também referenciado por *expected HSP length* ou tamanho de ajuste, é dado pela seguinte fórmula:

$$I = \ln (Kmn)/\lambda \qquad \text{[Equação 4]}$$

O parâmetro *K* é a mesma constante referente às probabilidades dos pares formados. O parâmetro λ se refere à constante da tabela de pontuação, assim como na Equação 1.

```
  Database: /home/local/wublast/discoLocal/nt.023
    Posted date:  Jul 8, 2007  7:37 PM
  Number of letters in database: 856,011,272
  Number of sequences in database:  216,838

Lambda     K      H
    1.37    0.711     1.31

Gapped
Lambda     K      H
    1.37    0.711     1.31


Matrix: blastn matrix:1 -3
Gap Penalties: Existence: 5, Extension: 2
Number of Hits to DB: 7,228,715
Number of Sequences: 5214551
Number of extensions: 7228715
Number of successful extensions: 146972
Number of sequences better than 10.0: 85
Number of HSP's better than 10.0 without gapping: 84
Number of HSP's successfully gapped in prelim test: 1
Number of HSP's that attempted gapping in prelim test: 146672
Number of HSP's gapped (non-prelim): 301
length of query: 1124
length of database: 20,481,288,407
effective HSP length: 22
effective length of query: 539
effective length of database: 20,366,568,285
effective search space: 10977580305615
effective search space used: 10977580305615
T: 0
A: 0
X1: 11 (21.8 bits)
X2: 15 (29.7 bits)
S1: 12 (24.3 bits)
S2: 20 (40.1 bits)
```

Figura 2 – Exemplo das informações do rodapé de saída BLAST

Este valor I é subtraído do tamanho atual da seqüência de consulta e também subtraído de todas as seqüências do banco de dados, obtendo assim os tamanhos considerados efetivos. As fórmulas para cálculo do tamanho efetivo das seqüências, sendo *m'*, para a seqüência de consulta e *n'*, para as seqüências da base de dados, são:

$$m' = m - I \qquad \text{[Equação 5]}$$

$$n' = n - (I * \text{números\_seqüências\_banco\_de\_dados}) \qquad \text{[Equação 6]}$$

Atualmente existem duas implementações principais para o algoritmo BLAST, NCBI BLAST[8] e WuBLAST[14]. Para a implementação WuBLAST, além do e-value ser informado, o *p-value* também é.

Dado um HSP com *score* S, o *p-value* é a probabilidade de outra seqüência aleatórias de mesmo tamanho e composição de letras obter o mesmo *score*. Os valores *p-value* e *e-value* são similares e de forma diferentes representam a mesma coisa. A diferença básica é que o primeiro dá a probabilidade de se obter uma pontuação (*score*) por pura aleatoriedade, enquanto o segundo fornece a probabilidade de ser encontrado um *hit* específico. A partir do *e-value*, o *p-evalue* pode ser obtido da seguinte forma:

$$P(S) = 1 - e^{-E(S)} \qquad \text{[Equação 7]}$$

3 Estatísticas em Bancos de Dados Fragmentados

A partir da década de 90, quando a Ferramenta BLAST passou a ser utilizada para atividades de comparar bioseqüências, já se imaginou a possibilidade de realizar esta tarefa de forma distribuída e/ou paralela.

Inicialmente foram adotados procedimentos simples, como replicar a base de dados em várias máquinas e compartilhar as seqüências de consulta entre elas [9, 6]. Contudo, alguns trabalhos[1,3] mostraram que ao invés de replicado, o banco de dados deveria ser fragmentado em máquinas distintas, com a economia no acesso ao disco trazendo ganhos no tempo total de execução. Muitas referências passaram então a fragmentar a base de dados de diversas formas, mostrando os ganhos obtidos sobre a estratégia fragmentada [1,3]. Ao final do processamento da estratégia fragmentada, os relatórios gerados são montados para que sejam semelhantes aos resultados da execução serial convencional.

É exatamente neste momento que pode haver uma dificuldade na validação do relatório de saída. Pois ao juntar os vários resultados de saída referente aos diversos fragmentos, deve-se fazer os ajustes para que os dados estatísticos se baseiem em todo o banco de dados e não somente no fragmento utilizado. Com a estratégia das bases fragmentadas o espaço de busca efetivo (Equações 4, 5 e 6) passa a ser diferente.

Mostramos no decorrer deste texto alguns testes comparando diferentes seqüências de consulta com vários fragmentos da base de dados, a fim de verificar os valores estatísticos fornecidos pela ferramenta BLAST. Para este testes utilizamos a mesma base de dado nr (obtida em [8]) dividida em diversos fragmentos. As seqüências de consulta variaram de acordo com a necessidade dos testes. Os *hits* utilizados foram escolhidos aleatoriamente, considerando que o mesmo *hit* estivesse localizado na comparação com todos os fragmentos para que os valores pudessem ser comparados.

A Tabela 1a mostra a variação do *e-value* para alguns *hits* aleatórios, para fragmentos de diferentes tamanhos da mesma base de dados. A segunda coluna mostra os valores

para o banco de dados inteiro, ou seja o *e-value* original, e nas outras colunas os valores do *e-value* para fragmentos desta base de dados. Cabe observar que a medida que o fragmento diminui de tamanho, mais o e-value se distancia do valor original, conforme Equação 1.

Seqüências da Base de Dados	*nr*	*nr/2*	*nr/4*	*nr/8*	*nr/24*	*nr/48*
hit1	2e-53	1e-53	6e-54	3e-54	1e-54	6e-55
hit2	5e-51	2e-51	1e-51	7e-52	2e-52	1e-52
hit3	2e-45	1e-45	6e-46	3e-46	1e-46	6e-47
hit4	1e-25	6e-26	3e-26	2e-26	6e-27	3e-27
hit5	1e-09	6e-10	3e-10	2e-10	6e-11	3e-11
hit6	5e-08	3e-08	1e-08	7e-09	3e-09	1e-09

Tabela 1a - E-values obtido sem a utilização de parâmetros.

Na Tabela 1b é possível visualizar os dados estatísticos obtidos a partir das equações da Seção anterior. Para estes dados foram utilizados os parâmetros padrões da BLAST, uma seqüência de consulta de aproximadamente 1,000 caracteres e variações no tamanho da base de dados nr[8]. Como pode ser observado, de acordo com que o banco de dados diminui, menor é o tamanho de ajuste. Como já havíamos mostrado nas Equações 5, 6 e 7, o tamanho do espaço de busca efetivo depende do tamanho de todo o banco de dados, o que pode ser visto na Tabela 1b.

Base de Dados	**Número de Seqüências**	**Tamanho da Base de Dados**	**Tamanho de ajuste**	**Tamanho efetivo da seqüência de consulta**	**Tamanho efetivo da Base de Dados**	**Tamanho efetivo do espaço de busca**
nr	4,874,565	1,684,337,227	143	906	987,274,432	894,470,635,392
nr1/2	2,501,434	842,168,703	138	911	496,970,811	452,740,408,821
nr1/4	1,220,850	421,085,148	133	916	258,712,098	236,980,281,768
nr1/8	578,282	210,542,430	129	920	135,944,052	125,068,527,840
nr1/24	188,161	70,180,789	121	928	47,413,308	43,999,549,824
nr1/48	100,119	35,090,166	116	933	23,476,362	21,903,445,746

Tabela 1b – Dados estatísticos obtidos a partir da Ferramenta NCBI BLAST.

O desafio para este ajuste de valores é possibilitar que, ao executar o BLAST utilizando uma estratégia onde a base de dados esteja fragmentada, o resultado de saída BLAST seja semelhante com as informações obtidas quando é utilizado todo o banco de dados, e não apenas um fragmento.

4 Parâmetros BLAST

Uma solução para fornecer informações de toda a base quando se utiliza somente um fragmento, seria a utilização de alguns parâmetros fornecidos pelas implementações NCBI-BLAST[8] e WuBLAST[14]. Para este propósito dois parâmetros, "*z*" e "*Y*", estão disponíveis.

4.1. Parâmetro *z*

O parâmetro *z* permite que seja informado o tamanho em caracteres das seqüências do banco de dados. Desta forma é possível executar uma seqüência contra um fragmento da base de dados informando o tamanho total da base, esperando-se que o *e-value* original seja obtido [10]. Contudo, ao analisarmos os resultados, o uso deste parâmetro também não permite obter os mesmos valores. A Tabela 2a mostra os valores obtidos para o *e-value* quanto utilizamos o parâmetro *z* com o tamanho correto da base. Pode-se perceber a pequena diferença na medida em que a base diminui de tamanho. Esta diferença ocorre devido ao cálculo do tamanho de ajuste ainda ser diferente quando utilizado todo o banco de dados, vide a coluna de Tamanho de Ajuste na Tabela 2b. O parâmetro "z" somente informa o tamanho atual da base. Com isso, valores como K e λ (necessários para Equações 4) ainda se baseiam no fragmento da base de dados.

Na Tabela 2b é possível observar os valores estatísticos obtidos com a utilização do parâmetro *z*. Como se vê, os valores estão próximos aos originais, mas ainda não são iguais. Para este teste mantivemos as mesmas seqüências de consulta e da base de dados das Tabelas 1a e 1b.

Seqüências da Base de Dados	nr	nr/2	nr/4	nr/8	nr/24	nr/48
hit1	2e-53	3e-53	4e-53	4e-53	4e-53	4e-53
hit2	5e-51	6e-51	7e-51	8e-51	8e-51	8e-51
hit3	2e-45	3e-45	4e-45	4e-45	4e-45	4e-45
hit4	1e-25	2e-25	2e-25	2e-25	2e-25	2e-25
hit5	1e-09	2e-09	2e-09	2e-09	2e-09	2e-09
hit6	5e-08	7e-08	8e-08	8e-08	9e-08	9e-08

Tabela 2a - E-values obtidos com a utilização do parâmetro *z*.

Uma estratégia para obter valores originais, como se a base não fosse fragmentada, seria colocar no parâmetro *z* o tamanho efetivo da base de dados. Assim verificaríamos a influência que tem o ajuste obtido a partir da Equação 5. Os resultados para esta es-

tratégia estão na Tabela 3a e 3b. Também para este teste mantivemos as mesmas seqüências de consulta e da base de dados que na Tabela 1a e 1b.

Base de Dados	Número de Seqüências	Tamanho da Base de Dados	Tamanho de ajuste	Tamanho efetivo da seqüência de consulta	Tamanho efetivo da Base de Dados	Tamanho efetivo do espaço de busca
nr	4874565	1,684,337,227	143	906	987,274,432	894,470,635,392
nr1/2	2501434	1,684,337,227	145	904	1,321,629,297	1,194,752,884,488
nr1/4	1220850	1,684,337,227	146	903	1,506,093,127	1,360,002,093,681
nr1/8	578282	1,684,337,227	146	903	1,599,908,055	1,444,716,973,665
nr1/24	188161	1,684,337,227	146	903	1,656,865,721	1,496,149,746,063
nr1/48	100119	1,684,337,227	146	903	1,669,719,853	1,507,757,027,259

Tabela 2b - Dados estatísticos obtidos com NCBI BLAST e uso do parâmetro *z*.

Na Tabela 3b são listados os valores estatísticos obtidos com o parâmetro *z* informando-se o tamanho efetivo da base de dados. Uma importante observação que se pode obter destas Tabelas é que, no caso da Tabela 3a, quanto menor a base de dados, mais próximo é o *e-value* obtido em relação ao original. Para perceber isso basta observar as últimas colunas da Tabela 2a e 3a, e compará-las à segunda coluna da Tabela 1a.

Seqüências da Base de Dados	nr	nr/2	nr/4	nr/8	nr/24	nr/48
hit1	8e-54	2e-53	2e-53	2e-53	2e-53	2e-53
hit2	2e-51	3e-51	4e-51	4e-51	5e-51	5e-51
hit3	8e-46	2e-45	2e-45	2e-45	2e-45	2e-45
hit4	4e-26	8e-26	1e-25	1e-25	1e-25	1e-25
hit5	4e-10	8e-10	1e-09	1e-09	1e-09	1e-09
hit6	2e-08	3e-08	4e-08	5e-08	5e-08	5e-08

Tabela 3a - E-values obtidos com parâmetro "z" e o tamanho efetivo do Banco de Dados.

Na Tabela 3a, quanto menor a base de dados, mais próximo é o *e-value* em relação ao original. Para compreender as razões, pode-se observar o Tamanho de Ajuste na Tabela 3b. Este também segue a mesma curva, onde quanto menor o tamanho da base, mais tende-se aproximar do Tamanho de Ajuste original. Isto pode ser explicado, pois na medida em que o tamanho do fragmento diminui, os valores da Equação 4 também diminuem, exceto o valor de *m*, passado como parâmetro. Como *m* permanece constante, o valor *m'*, resultante da Equação 5, tende a aumentar. Contudo o *m'* aumentará no máximo até o valor passado pelo parâmetro *z*. Como o tamanho efetivo da base foi passado por parâmetro, quanto menor a base de dados mais próximo do original será o

e-value obtido. Desta forma, utilizando-se o parâmetro "*z*", percebemos melhores resultados do que quando os valores reais da base de dados são utilizados.

Base de Dados	Número de Seqüências	Tamanho da Base de Dados	Tamanho de Ajuste	Tamanho efetivo da seqüência de consulta	Tamanho efetivo da Base de Dados	Tamanho efetivo do espaço de busca
nr	4,874,565	987,274,432	135	914	329,208,157	300,896,255,498
nr1/2	2,501,434	987,274,432	140	909	637,073,672	579,099,967,848
nr1/4	1,220,850	987,274,432	141	908	815,134,582	740,142,200,456
nr1/8	578,282	987,274,432	142	907	905,158,388	820,978,657,916
nr1/24	188,161	987,274,432	142	907	960,555,570	871,223,901,990
nr1/48	100,119	987,274,432	142	907	973,057,534	882,563,183,338

Tabela 3b – Valores estatísticos obtidos a partir da Ferramenta NCBI BLAST, com uso do parâmetro "z", passando o tamanho efetivo da Base de Dados.

4.2. Parâmetro *Y*

Outro parâmetro também utilizado, mas desta vez somente na Ferramenta NCBI BLAST é o *Y*. Com ele é possível informar diretamente o tamanho efetivo do espaço de busca da pesquisa, e evitar que o resultado da Equação 5, utilizando fragmentos, seja considerado. Mas para isso é importante que já se tenha contabilizado o tamanho efetivo do espaço de busca, sendo necessário tanto a seqüência de consulta como toda a base de dados .

Seqüências da Base de Dados	nr	nr/2	nr/4	nr/8	nr/24	nr/48
hit1	2e-53	2e-53	2e-53	2e-53	2e-53	2e-53
hit2	5e-51	5e-51	5e-51	5e-51	5e-51	5e-51
hit3	2e-45	2e-45	2e-45	2e-45	2e-45	2e-45
hit4	1e-25	1e-25	1e-25	1e-25	1e-25	1e-25
hit5	1e-09	1e-09	1e-09	1e-09	1e-09	1e-09
hit6	5e-08	5e-08	5e-08	5e-08	5e-08	5e-08

Tabela 4a - E-values obtidos com a utilização do parâmetro Y.

Como pode ser visto na Tabela 4a, o uso do parâmetro *Y* que informe o espaço de busca efetivo pode ser uma forma segura de se conseguir *e-values* iguais aos originalmente obtidos pelo BLAST com a base não fragmentada. Para todas as seqüências, as colunas relativas ao BLAST com bancos fragmentados obtiveram os mesmos valores para o banco de dados completo.

De acordo com a Tabela 4b, podemos verificar que mesmo com valores estatísticos distintos nos bancos fragmentados, como é o caso do tamanho de ajuste, os e-values

permaneceram idênticos aos valores originais. Isto ocorre pois na Equação 3 o espaço de busca é definido por parâmetro de entrada.

Base de Dados	Número de Seqüências	Tamanho da Base de Dados	Tamanho de ajuste	Tamanho efetivo da seqüência de consulta	Tamanho efetivo da Base de Dados	Tamanho efetivo do espaço de busca
nr	4,874,565	1,684,337,227	143	906	987,274,432	894,470,635,392
nr1/2	2,501,434	842,168,703	138	911	496,970,811	894,470,635,392
nr1/4	1,220,850	421,085,148	133	916	258,712,098	894,470,635,392
nr1/8	578,282	210,542,430	129	920	135,944,052	894,470,635,392
nr1/24	188,161	70,180,789	121	928	47,413,308	894,470,635,392
nr1/48	100,119	35,090,166	116	933	23,476,362	894,470,635,392

Tabela 4b - Valores obtidos com a Ferramenta NCBI BLAST, com uso do parâmetro Y.

O que podemos concluir com estas tabelas é que fornecer valores utilizando dados efetivos já calculados quando utilizamos o banco de dados completo(não fragmentado) é sem dúvida alguma uma boa opção, principalmente utilizando-se o parâmetro *Y*. Contudo, obter os tamanhos efetivos não é nada trivial, já que é difícil obter valores de ajuste que não os fornecidos pela própria ferramenta BLAST.

4.3. Cálculos Prévios

Como alternativa à passagem de parâmetros, a ferramenta mpiBLAST[1] opta por uma estratégia diferente. São feitos os cálculos necessários utilizando as funções disponíveis dentro do NCBI BLAST e enviados os resultados entre os diversos nós para que todos os fragmentos retornem resultados exatos aos valores originais.

Antes de prosseguir, cabe observar que muitos dos conceitos aqui discutidos foram tirados do código fonte das ferramentas, assim como de emails trocados com os próprios desenvolvedores.

O mpiBLAST não necessita conhecer os modelos de pontuação para definição dos valores de K e λ pois ele acessa as funções disponibilizadas na própria ferramenta BLAST e, assim, obtém os espaços de busca efetivos para cada seqüência de consulta. Para este acesso às funções é disponibilizado um *patch*[3] que altera o código fonte da ferramenta BLAST. Dentre outras coisas, o *patch* permite ao mpiBLAST optar por chamar as funções para geração de dados estatísticos ou para comparação de seqüências pelas estações de trabalho.

O mpiBLAST obtém de forma serial o tamanho efetivo da seqüência de consulta e da base de dados, fazendo chamadas às funções para filtrar as seqüências e calcular o

3 Patch é um programa de computador que aplica as diferenças textuais entre dois programas e, mais freqüentemente, a arquivos de computador contendo essas diferenças. Uma vez que você tem alguma das versões dos elementos e o patch, você consegue transformar uma na outra, e vice versa.

espaço de busca sem necessariamente desempenhar a busca de similaridade. Além deste processo ser executado para todas as seqüências de consulta, ele é dito serial pois a princípio a base de dados está totalmente fragmentada e localizada somente na máquina reconhecida pelo processo inicial. O custo para cálculo do tamanho efetivo para cada seqüência de consulta pode consumir muito tempo de processamento.

Como os próprios desenvolvedores do mpiBLAST reconhecem que este tempo de processamento inicial pode ser muito grande, foi adicionada a opção de se obter *e-values* não tão exatos, porém com um tempo de execução muito menor. O espaço de busca efetivo passa a ser obtido somente para uma seqüência de consulta, e este resultado é que será enviado para todas a máquinas de trabalho utilizarem como referencia no processamento das outras seqüências. Embora o resultado não seja o mesmo, os desenvolvedores comentam que não são geradas diferenças significativas, a não ser que a seqüência de consulta utilizada seja muito distinta do padrão de tamanho e conteúdo das outras seqüência de consulta.

4.4. Fragmentação Virtual

Diferentemente das estratégias de execução paralela do BLAST até aqui comentadas, outra abordagem seria a fragmentação virtual, em que o banco de dados completo é fragmentado somente logicamente. Desta forma, cada máquina de trabalho utiliza somente um segmento da base de dados, sem a necessidade da pré-formatação. Conseqüentemente, os valores necessários para cálculos estatísticos estariam sempre disponíveis para a execução do BLAST, embora este utilize somente uma parte da base de seqüências.

Como exemplo, podemos citar uma solução adotada com a ferramenta WuBLAST, que permite que através de parâmetros explícitos seja informado o intervalo da base de dados que se deseja comparar com a seqüência de consulta. Desta forma não é necessário gerar fragmentos físicos da base de dados e não há necessidade de ajuste do valor *e-value*.

Outra opção seria utilizar programas que permitam o compartilhamento de parte dos dados a partir de uma fonte em uma única máquina. É o caso da ferramenta Global Arrays [7], que facilita a distribuição dos dados a partir de uma interface de programação que pode ser utilizada por arquiteturas de memória compartilhada ou distribuída. Para este caso, a estratégia [2] de BLAST paralelo consegue usufruir de fragmentos virtuais, escapando dos tratamentos de valores estatísticos. Cabe observar que trata-se de uma solução intrusiva, pois houve necessidade de alteração do código fonte BLAST para que as leituras da base de dados sejam feitas na ferramenta Global Arrays.

4.5. Discussão

Diante das propostas de correções para o *e-value* quanto se utiliza os bancos fragmentados, a primeira decisão a ser tomada é se a solução será ou não intrusiva ao código BLAST. Caso a resposta seja afirmativa, as funções referentes às estatísticas de alinhamento podem ser alteradas, passando-se a considerar o acesso à base de dados fragmentada. Entretanto, em soluções intrusivas existe a dificuldade em fazer uso de futuras atualizações do BLAST, além disso, impossibilita que outras ferramentas de alinhamento sejam utilizadas.

Em estratégias não intrusivas, uma boa opção seria o uso do parâmetro *z* com o tamanho efetivo da base de dados. Este valor pode ser obtido no campo *effective length of database*, no rodapé de saída do BLAST (Figura 2). É necessário porém, que esta saída seja o resultado da comparação com o banco de dados completo.

Na Figura 3a, para uma seqüência de consulta com cerca de 1,000 caracteres (seq1), mostra-se a variação do *e-value* para vários *hits* na medida em que o fragmento da base de dados diminui. Nesta figura utilizamos com o parâmetro *z* o tamanho efetivo da base de dados. A fim de analisar o resultado desta estratégia, vamos utilizar outras seqüências de consulta, mantendo o mesmo valor do parâmetro *z* obtido com a seqüência seq1.

Para este testes utilizamos duas seqüências de consulta com tamanhos bem distintos, uma com quase 300 caracteres (seq2), e outra com cerca de 10,000 caracteres (seq3). O resultado para estas duas consultas pode ser visto na Tabela 5. O *hit1* se refere a um *hit* qualquer quando a seqüência seq2 é considera. Para este *hit,* o valor do *e-value* quando comparado a todo banco de dado sem o uso do parâmetro é 2e-50.

De forma análoga, no caso do *hit2*, outro *hit* arbitrário escolhido, mas neste caso com seqüência seq3. O valor do e-value sem o uso de parâmetros para a base de dados completa é de 2e-70.

Seqüências da Base de Dados	**nr**	**nr/2**	**nr/4**	**nr/8**	**nr/24**	**nr/48**
hit1	7e-51	1e-50	2e-50	2e-50	2e-50	2e-50
hit2	7e-71	2e-70	2e-70	2e-70	3E-70	3e-70

Tabela 5: Exemplo de hits com diferentes seqüências de consulta para mesmo valor do parâmetro z

Pode-se observar para estes dois testes que o *e-value* obtido é muito próximo do original. Embora aqui tenhamos mostrado somente alguns dos vários testes feitos, o uso desta estratégia pode ser uma boa estratégia quando utiliza-se fragmentação da base de dados para execuções paralelas da ferramenta BLAST.

5 Conclusões e Trabalhos Futuros

Nosso objetivo com este trabalho foi mostrar o comportamento do *e-value* quando a ferramenta BLAST é executada com fragmentos da base de dados. Para isso foram analisados os diversos resultados com uso variado de parâmetros dos programas. Comenta-se também eventuais alterações no código BLAST, com uso direto das funções de estatísticas de alinhamento.

Foi notado que o uso do parâmetro *z* possibilita *e-values* próximos do desejado, principalmente quando valores efetivos da base de dados são utilizados. Já o uso do parâmetro *Y* é outra boa opção, mas exige que sejam feitos cálculos para o espaço de busca efetivo antes que a comparação ocorra. Ferramentas, como mpiBLAST, possibili-

tam que os cálculos para uso do parâmetro *Y* sejam bem precisos, contudo exigem em contra partida muito tempo de processamento.

Obter *e-values* próximos dos valores de todo o banco de dados quanto se utiliza fragmentos não é uma tarefa trivial. Enquanto alguns pesquisadores exigem idênticos *e-value*, semelhantes aos originais, outros toleram pequenas diferenças e valores aproximados. Muitos pesquisadores dizem que a diferença obtida com o parâmetro *z* não é significativa, podendo variar dentro de uma taxa de 5% do valor original sem nenhuma perda de precisão [5].

Com os testes feitos, alguns destes aqui apresentados, sugere-se que o melhor opção para manter a precisão do *e-value* em bases de dados fragmentadas é a utilização do parâmetro *z* com o tamanho efetivo da base de dados.

Como trabalhos futuros, acreditamos interessante soluções que utilizem a fragmentação virtual dos dados sem alteração no código fonte da ferramenta. Desta forma, fragmentos lógicos da base de dados seriam processados, mas tendo disponível toda a base de dados e mantendo correto os valores estatísticos.

Outra opção seria o procedimento de incorporar precisão nas estatísticas de alinhamentos enquanto a base de dados está sendo atualizada durante o processamento. Pois, em situações como estas, o ideal é o sistema incorporar a atualização da base de dados aproveitando todo o processamento já ocorrido, e proceder com o restante da comparação contra a base de dados atualizada. Contudo o problema aqui está no ato de proceder com o restante da comparação, pois com a base de dados alterada o valor que define o *e-value* é dependente da banco de dados alterado.

Referências

[1] A. E. Darling, L. Carey, W. Feng; The Design, Implementation, and Evaluation of mpiBLAST;ClusterWord Conference & Expo and the 4th International conference on Linux Clusters: The HP Revolution 2003; LA-UR 03-2862, 2003;

[2] C. Oehmen, J. Nieplocha; ScalaBLAST: A Scalable Implementation of BLAST for High-Performance Data-Intensive Bioinformatics Analysis; IEEE Transactions on Parallel and Distributed Systems, v.17, p.740-749, 2006;

[3] D. Mathog; Parallel BLAST on split database; Bioinformatics, v.19, p.1864-1866, 2003;

[4] I. Korf, M. Yandell, J. Bedell;An Essential Guide to the Basic Local Alignment Search Tool; O' Reilly & Associates, Inc., Sebastopol, U.S.A.,2003;

[5] I. Rossi, D. Machignoli, D. Medini, R. Beltrami, C. Donati, P. Fariselli, A. Covacci, R. Casadio; A simple yet effective implementation of a parallel BLAST for computer clusters; Virtual Conference on Genomics and Bioinformatics Papers, v.1, p.10-14, 2002;

[6] J. D. Grant, R. L. Dunbrack, F. J. Manion, M. F. Ochs; BEO-BLAST: Distributed BLAST and PSIBLAST on Beowulf Cluster; Bioinformatics Applications Note, v.18, p.765-766, 2002;

[7] J. Nieplocha, R. Horrison, R. Littlefield; Global Arrays: a Nonuniform Memory Access Programming Model for High-Performance Computers; Journal Supercomputing, v.10, p.197-220, 1996;

[8] NCBI BLAST; Disponível em: http://www.ncbi.nlm.nih.gov/BLAST; Acesso em 18 jul.2007;

[9] O. T. Salazar, E. L. Zapata, J. M. Carazo; On a efficient parallelization of exhaustive sequence comparison algorithms on messege passing architectures; Bioinformatics, v. 8, p.765-766, 2002;

[10] R. Costa, S. Lifschitz; Database allocation strategies for parallel blast evaluation on clusters; Distributed and Parallel Databases, v.13, p.99-127, 2003;

[11] S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman; Basic Local Alignmen-Search Tool; Journal Molecular Biology, v.215, p.403-410, 1990;

[12] S. Altschul, W. Gish; Local alignment statistics; Methods Enzymol., v.266, p.460–480, 1996;

[13] S. Altschul, W. Gish; The estimation of statistical parameters for local alignment score distributions; Nucleic Acids Research, v.29, p.351-361, 2001;

[14] WU-BLAST; Disponível em: http://blast.wustledu/; Acesso em: 18 jul.2007.