# 7
# Conclusão e trabalhos futuros

O problema de processamento paralelo para comparação de seqüências vêm sendo há algum tempo estudado na biologia computacional. A utilização da ferramenta BLAST em um agrupamento de computadores tem sido implementada por diversos trabalhos, principalmente com o objetivo de obter ganhos de desempenho.

## 7.1
## Resumo

Entre os vários trabalhos de paralelização da execução da Ferramenta BLAST, duas abordagens de alocação da base de dados possuem maior destaque, são elas: Replicada e Fragmentada. Na primeira, cada estação de trabalho recebe a base de dados e somente algumas das seqüências de consulta para processamento, enquanto que, na abordagem Fragmentada, cada estação de trabalho recebe todas as seqüências de consulta e somente alguns fragmentos da base de dados.

Dos trabalhos comentados no Capítulo 4, os mais atuais buscam obter balanceamento de carga partindo do pressuposto que a estratégia seja fragmentada e, de acordo com o processamento alguns fragmentos, são replicados para algumas máquinas, no intuito de utilizar recursos ociosos. Nossas propostas partem do pressuposto contrário. Isto é, a princípio toda a base de dados está replicada, mas de tal forma que cada réplica está dividida em vários fragmentos distintos. Inicialmente consideramos uma base de dados replicada e ao mesmo tempo fragmentada em cada máquina de trabalho, contudo admitimos que a replicação não é uma restrição das nossas estratégias, caso uma máquina não tenha espaço em disco suficiente. Esta abordagem permitiu implementar um balanceamento de carga não só na submissão de seqüências de consulta, mas também nos fragmentos da base de dados.

A partir dos estudos dos fatores de desvio de tempo (desbalanceamento) durante o processamento paralelo da ferramenta BLAST, sejam eles intrínsecos ou não, sugerimos duas estratégias de execução com balanceamento de carga em ambientes paralelos. A primeira divide toda a tarefa e as submete sempre que uma estação estiver ociosa, chamada de Sob Demanda. A segunda estratégia divide toda a tarefa de forma igual entre as máquinas de trabalho, e re-aloca pequenas tarefas à medida que as máquinas estejam ociosas, essa é chamada de Estratégia Corretiva.

Segundo os testes feitos verificamos um bom balanceamento de carga em ambas as estratégias, principalmente considerando o desvio de tempo causado pelo tamanho da seqüência de consulta ou similaridade da mesma. Embora não tenhamos feito testes em ambiente com processos concorrentes, o balanceamento obtido com os fatores de desvio de similaridade e tamanho justificam as propostas.

Além do balanceamento, verificamos que as abordagens de alocação da base de dados facilitaram o ajuste na implementação para diminuir o tempo final de execução do BLAST em ambientes paralelos. Permitindo, ainda, uma aceleração linear à medida que novas máquinas fossem integradas ao ambiente de execução.

Importante comentar que as nossas propostas agem de forma não intrusiva ao código fonte da ferramenta BLAST, sendo totalmente adaptável a qualquer atualização ou mesmo outra ferramenta que produza alinhamentos entre biosseqüências.



## 7.2 Contribuições

Após a análise dos requisitos necessários à execução da ferramenta BLAST em ambientes paralelos nos Capítulo 2 e 4, podemos listar as contribuições desta dissertação da seguinte forma:

- Estudo dos fatores de desvio de balanceamento intrínseco à ferramenta BLAST;
- Estado da arte às estratégias de execução da ferramenta BLAST em ambientes paralelos;
- Implementação de alocação da base de dados com alocação replicada e fragmentos primários;

– Sugestões de duas estratégias de submissão de seqüências de consulta, que obtenham balanceamento de carga durante o processamento BLAST paralelo;
– Disponibilização de ferramentas que possam ser obtidas gratuitamente para execução em agrupamentos proprietários, assim como os procedimentos de instalação;
– Sugestões de estratégias de execução robustas aos problemas ocorridos durante o processamento paralelo.

## 7.3 Trabalhos Futuros

Além destas contribuições, sugerimos alguns trabalhos futuros. A começar por uma análise crítica, que possa comparar ambas as estratégias: Sob Demanda e Corretiva. Permitindo que estudos possam dizer em qual ambiente cada estratégia terá melhor desempenho, ou mesmo qual deverá ser utilizada em ambientes de execução exclusivo ou não. Adicionado a isso, estudos que possam sugerir parâmetros precisos do uso dessas estratégias.

Outro trabalho futuro é a adaptação de um método de acesso à base de dados. O mesmo evita o pré-processamento de dividir a base de dados, e ao mesmo tempo, mantém a paralelização de acesso ao disco. Isto pode ser obtido através de estratégias virtualmente fragmentadas, acessando partições lógicas de toda a base de dados. Um exemplo disso é o trabalho de (24), que permite o BLAST comparar uma seqüência de consulta contra um banco de dados completo de forma serial, tratando toda a base em blocos não divididos fisicamente.

Outra sugestão de trabalho futuro é a utilização de *drivers* para gerência de acesso aos dados locais, provendo as seqüências da base ao BLAST de forma otimizada. Esta estratégia de disponibilização dos dados já tem sido implementada de forma seqüencial por (24), e se mostra interessante em casos que várias seqüências são processadas em uma única tarefa. Desta forma, no caso da estratégia Sob Demanda, as seqüências serão processadas concorrentemente.

Diante das estratégias de processamento paralelo sugeridas, em que uma tarefa é dividida em diversas outras, acreditamos que nossas contribuições possam ser estendidas à ambientes de grades computacionais, utilizando grande quantidade de máquinas, e reajustando as propriedades do balanceamento de carga.

Mesmo considerando que em nossas sugestões existe a participação de uma estação de gerência, acreditamos que alterações possam ser feitas no intuito de se adaptar às configurações $peer - to - peer$. No caso da estratégia Corretiva talvez seja necessário alterações no módulo de re-alocação das tarefas, incluindo heurísticas de balanceamento de carga.

Uma desvantagem na implementação da alocação da base de dados aqui proposta é a necessidade de armazenar em cada estação de trabalho uma réplica do banco de dados com seus fragmentos. Sugerimos então como trabalho futuro, a possibilidade em alocarmos somente alguns fragmentos e não toda a base, fazendo uso do espaço disponível, mesmo que onerando o custo de processamento. Para isso, abordagens com redundância mínimas e estratégias inteligentes poderão ser utilizadas.

Outra proposta de trabalho futuro é de um estudo mais aprofundado referente à taxa de processamento enviado por unidade de trabalho. Em nosso caso isso se aplica à quantidade de seqüências enviadas por tarefa (Figura 5.8) e quantidade de fragmentos de uma base de dados (Figura 6.6). Sugerimos que estudos sejam feitos no intuito de verificar se a dosagem de trabalho pode depender ou não das configurações das máquinas utilizadas, ou de alguns parâmetros de teste. Talvez estudos com abordagens mais formais possam reduzir a um número ótimo a taxa de processamento.