

# CONTROLE DE MEMÓRIA PARA PROCESSOS BLAST

**Aluno: Diogo Gitahy**  
**Orientador: Sérgio Lifschitz**

## Introdução

Foi estudada uma forma de diminuir os custos de execução de processos Blast. A estratégia utilizada consistia em controlar a disponibilização de memória para os processos correntes, controlando sua execução de forma a otimizá-la.

## Objetivos

Otimizar a execução de processos Blast diminuindo o tamanho de memória necessária à sua execução e/ou diminuindo o número de acessos à memória para um ou mais processos.

## Metodologia

Para utilização da ferramenta BioProvider [1] é necessária a instalação do driver e a formatação dos bancos de dados a partir dos bancos no formato fasta. Tais procedimentos necessitam de informações variáveis de acordo com o interesse do usuário no momento da instalação.

A fim de tornar mais prática essa rotina de configuração, foram criados três scripts: o primeiro para formatar o banco de dados, o segundo para instalar o driver e o último para desinstalar o driver e remover apenas o necessário, visto que é interessante continuar com o banco formatado disponível em disco, devido a seu alto custo.

Esses scripts foram feitos em BASH, um tipo de *shell script*, visando a facilidade de configuração do sistema que ele dispõe. A única restrição é a necessidade de ter privilégios de root.

O script de formatação consiste em:

- obter os parâmetros passados pelo usuário em arquivo texto;
- dividir o banco no número de blocos necessário, unir esses blocos da forma correta e formatar cada resultado dessa união usando o *formatdb* do Blast;
- remover os blocos, já que já foram usados e não são mais necessários.

A rotina seguida pelo script de instalação se resume a:

- obter os mesmos parâmetros da configuração;
- criar os special files requeridos;
- instalar o driver.

Os passos de desinstalação são:

- obter os mesmos parâmetros passados para a configuração e a instalação;
- remover os special files;
- desinstalar o driver.

A execução do Blast em si, utilizando o BioProvider, foi deixada para o usuário, visto que não dependem de privilégios de root. Além disso o usuário pode criar um script para executar vários processos concorrentes.

Com os scripts criados, foram feitos scripts de teste para verificar a maior eficiência do Blast executado com a gerência de memória frente à execução sem o artifício.

Esse script consiste em variar o tamanho do anel e o banco de dados e para cada conjunto de parâmetros:

- executar o script de formatação;
- executar o blast sem o driver, medindo o tempo gasto;
- executar o script de instalação do driver;
- executar o blast com o driver, medindo o tempo gasto;
- executar o script de desinstalação;

Esse script é dividido em dois blocos: o primeiro executa o blast sem o driver com um banco e o blast com o driver com outro banco, enquanto o segundo faz o inverso. Isso evita que o resultado seja influenciado pelo fato de o banco ficar em memória.

O resultado da execução desse script indica o quão eficiente é a gerência de memória.

## **Conclusão**

Controlar o acesso e a disponibilização da memória é muito útil para otimizar os custos de qualquer programa, principalmente aqueles que utilizam bancos de dados. Saber ordenar os acessos e minimizar a quantidade desses diminuem os custos.

Além disso, ficou claro que a utilização de scripts para configurar o sistema, criar arquivos, executar programas e comandos do sistema é uma excelente opção. Basta saber criar scripts auxiliares que sirvam para testar pequenas partes do script macro, para que não seja necessário observar um erro no script final e procurar o local exato, o que é muito custoso.

## **Referências**

1 – FERREIRA DE NORONHA, Maíra . Controle da Execução e Disponibilização de Dados para Aplicativos sobre Seqüências Biológicas: o Caso BLAST. **Dissertação de Mestrado**, set 2006.