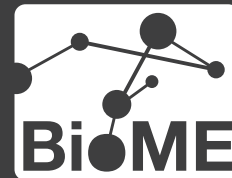


*In Company*

**CURSOS DE CURTA DURAÇÃO**

# **BIOINFORMÁTICA**

**BIOME - CENTRO MULTIUSUÁRIO DE BIOINFORMÁTICA - UFRN**



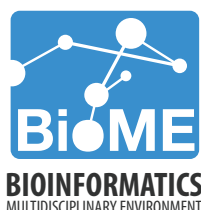
Curso teórico-prático

## **INTRODUÇÃO À ANÁLISE DE DADOS DE SEQUENCIADORES DE SEGUNDA GERAÇÃO**

Fiocruz/Biomanguinhos  
Rio de Janeiro - RJ

21 a 23 de Janeiro de 2020

## **Referencial de Aulas Práticas**



 [bioinfo.imd.ufrn.br](http://bioinfo.imd.ufrn.br)  
 Av. Odilon Gomes de Lima 1722  
Capim Macio, 59078-400  
Natal/RN - Brazil  
 [biome@imd.ufrn.br](mailto:biome@imd.ufrn.br)  
 +55 (84) 99480-6818  
+55 (84) 3342-2216 - Ramal 123

## ***Sobre o BioME***

O BioME (Centro Multidisciplinar de Bioinformática) é fruto de uma iniciativa em bioinformática da UFRN em Natal, Brasil. Ele foi criado no início de 2016 com a missão de promover a bioinformática no cenário regional e nacional, atuando em quatro diferentes níveis. No ensino, seus professores/pesquisadores atuam no nível de graduação, em disciplinas de bioinformática para diversos cursos na área de biociências, e na ênfase em Bioinformática do curso de Bacharelado em Tecnologia da Informação do Instituto Metrópole Digital (IMD). Adicionalmente, o BioME possui um programa de pós-graduação (PPg-Bioinfo), nível mestrado e doutorado, com conceito 5 na CAPES, que tem como objetivo formar recursos humanos de alto nível em bioinformática, tanto para a área acadêmica, como para atuação no setor produtivo/industrial. Na pesquisa, grupos multidisciplinares envolvidos com o BioME produzem ciência de ponta em bioinformática aplicada à diversas áreas como: biologia do câncer, modelagem de sistemas, biologia de sistemas, genômica, proteômica, evolução molecular, bioinformática estrutural, etc. No setor de prestação de serviços, um centro técnico multiusuário disponibiliza serviços de bioinformática e de análises de dados para clientes, tanto para grupos acadêmicos como para empresas do setor de biotecnologia. Por fim, o programa corporativo busca fomentar a interação produtiva com a indústria de biotecnologia, estendendo os conhecimentos produzidos na universidade para a sociedade.

## Sumário

Sobre o curso	01
Programa	02
Aula prática 1	03
Aula prática 2	12
Aula prática 3	20
Aula prática 4	30
Aula prática 5	39

## Sobre o curso

O curso teórico-prático “Análise de Dados de Sequenciadores de Nova Geração” pertence à grade de programação dos Cursos de Curta Duração em Bioinformática promovidos pelo BioME, com apoio do Programa de Pós Graduação em Bioinformática da UFRN e do Instituto de Bioinformática e Biotecnologia (2Bio) que, juntamente com cursos de outros temas da área de Bioinformática, recebeu mais de 280 alunos do Brasil e do exterior desde 2017.

Com carga horária total de 20h, o presente curso é um treinamento introdutório e direcionado, visando fornecer aos participantes uma sólida base para o início de análises de dados de sequenciadores de segunda geração, sendo ministrado pelos professores:



**Prof. Dr. Sandro J. de Souza**  
sandro@neuro.ufrn.br

Doutor em Bioquímica pela Universidade de São Paulo e Pew Latin American Fellow pela Universidade de Harvard -EUA, Dr. de Souza foi um dos pioneiros da área de Genômica e Bioinformática no Brasil. Foi membro associado do Ludwig Institute for Cancer Research, eleito pelo Forum Econômico Mundial como Young Global Leader em 2009, e professor visitante na Universidade de Chicago - EUA. Atualmente, é professor do Instituto do Cérebro da Universidade Federal do Rio Grande do Norte, onde é vice-coordenador do Programa de Pós-graduação em Bioinformática e diretor-fundador do Centro Multidisciplinar de Bioinformática (BioME). Mais informações: [Lattes](#)



**Prof. Dr. Jorge Estefano Santana de Souza**  
jorge@imd.ufrn.br

Graduado em Ciência da Computação e doutor em Bioinformática pela Universidade de São Paulo, o Prof. Jorge E.S. de Souza atuou como bioinformata no Ludwig Institute for Cancer Research, Recepta Biopharma, Hemocentro de Ribeirão Preto e AC Camargo Cancer Center. Atualmente é professor adjunto do Instituto Metrópole Digital da Universidade Federal do Rio Grande do Norte, onde é membro do Programa de Pós-graduação em Bioinformática e do Centro Multidisciplinar de Bioinformática (BioME). Tem experiência na área de Bioinformática e Genômica, atuando principalmente nos seguintes temas: Câncer, Biologia Molecular, Genômica e Transcriptômica. Mais informações: [Lattes](#)

## Programa

Data	Horário	Assunto
Segunda-feira 21/01/2020	08:30h	Apresentação
	08:45h	Introdução à Genômica e Bioinformática
	10:10h	Introdução ao Linux
	12:15h	Intervalo de almoço
	13:30h	Dados de NGS: Análise de Qualidade
	16:30h	Encerramento
Terça-feira 22/01/2020	08:30h	Dados de NGS: Chamada de Variantes
	12:15h	Intervalo de almoço
	13:30h	Dados de NGS: RNASeq mRNA
	16:30h	Encerramento
Quarta-feira 23/01/2020	08:30h	Dados de NGS: RNASeq ncRNA
	12:15h	Intervalo de almoço
	13:30h	Discussão de projetos trazidos pelos alunos e encerramento

## Aula prática 1

- Linux
- Explorando dados de NGS

**Professor:** Jorge Estefano Santana de Souza, [jorge@imd.ufrn.br](mailto:jorge@imd.ufrn.br);

### Objetivos:

Este tutorial tem como objetivo fazer com que o aluno conheça um pouco mais sobre o Linux, bem como habilitar o usuário para trabalhar com as principais ferramentas de bioinformática.

### Ferramentas:

- 1- SSH.
- 2- shell.
- 3- Comandos básicos do Linux.

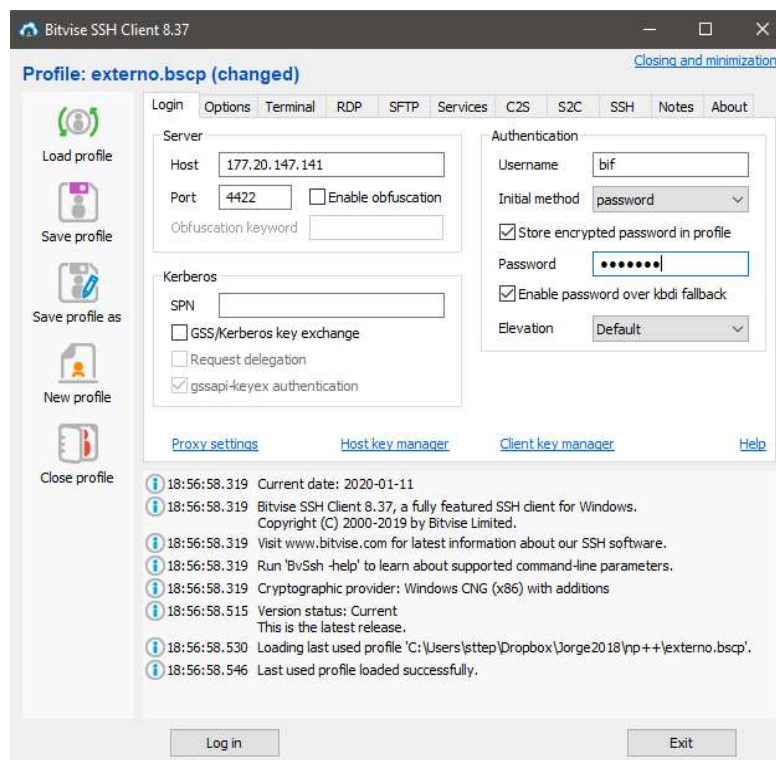
### Comandos Básicos:

Durante a execução do tutorial iremos abordar os comandos essenciais, no entanto é fortemente recomendável que os alunos expandam os seus conhecimentos aprendendo outros comandos básicos do Linux. Mais informação no site:

<http://wiki.ubuntu-br.org/ComandosBasicos>

### Login Servidor:

Inicialmente vamos fazer o logon no servidor abrindo um terminal na máquina remota:



```
User:  bif
Host:  177.20.147.141
Porta: 4422
Senha: bif0003
```

### No terminal LINUX (ou Mac):

Quando estiver conectado por ssh e quiser saber quem mais está trabalhando na máquina, use o comando **who**.

Como nos conectamos a várias máquinas com vários logins, as vezes precisamos digitar **whoami** para lembrar como nos conectamos.

Mais dois comandos precisam ser usados sempre:

ls - lista o conteúdo da pasta ou diretório;

pwd - mostra o caminho do presente diretório (path of working directory);

1) Vamos começar pelo básico, certifique-se de que a pasta atual é:

```
/data/home/bif
```

Para isso, digite o comando:

```
pwd
```

\*ps. vc vai ver onde está

2) Nesse momento pode não haver o que listar, mas sempre é importante ver o que tem, ou não tem, na pasta. Comando:

```
ls
```

3) Crie um diretório nomeando-o com o seu nome:

```
mkdir SeuNome
```

4) Digite ls:

```
ls
```

5) Entre no diretório com SeuNome:

```
cd SeuNome
```

com um pwd você deve ver /data/home/bif/SeuNome:

```
pwd
```

6) Suba de volta um diretório com `cd ..` - esses dois pontos direcionam para o diretório acima (change directory to upper directory):



```
cd ..
```

```
ls
```

Volte para seu diretório:

```
cd SeuNome
```

```
ls
```

### Outros comandos:

7) Ver o conteúdo do diretório: /home/treinamento:

```
ls /home/treinamento/
```

8) Copiar o arquivo lyrics para o diretório presente. Perceba o ponto que representa o presente diretório:

```
cp /home/treinamento/lyrics .
```

```
ls
```

9) Criar o diretório teste:

```
mkdir teste
```

```
ls
```

10) Mover o arquivo lyrics para o diretório teste:

```
mv lyrics teste
```

```
ls
```

11) Entrar no diretório teste:

```
cd teste
```

12) Trocar o nome do arquivo:

```
mv lyrics letra
```

```
ls
```

13) Copiar o arquivo:

```
cp letra lyrics
```

14) Remove o arquivo letra:

```
rm letra
```

15) Listar com mais informações:

```
ls -l
```

16) Mostrar o manual para o programa ls:

```
man ls
```

q (interrompe o output do manual)

17) Imprimir na tela o conteúdo do arquivo:

```
more lyrics
```

q (interrompe o comando more)

less também funciona para ver o conteúdo de qualquer arquivo:

```
less /home/treinamento/ERR844339.fastq
```

q(interrompe o comando more e lessdo Linux)

18) Imprimir as primeiras linhas do arquivo:

```
head /home/treinamento/ERR844339.fastq
```

19) Imprimir as últimas linhas do arquivo:

```
tail /home/treinamento/ERR844339.fastq
```

20) **tabulador(tab) e asterisco:** são usados para nomes compridos. O tabulador completa o nome, o asterisco funciona como coringa. Por exemplo **more ly\*** imprimirá o conteúdo de lyrics:

```
more ly*
```

21) Listar o conteúdo do diretório /home/treinamento/blast\_aula/FASTAS:

```
ls /home/treinamento/blast_aula/FASTAS/
```

22) Imprimir na tela o conteúdo do arquivo GAPDH:

```
more /home/treinamento/blast_aula/FASTAS/GAPDH
```

23) Copiar o arquivo GAPDH para o diretório atual:

```
cp /home/treinamento/blast_aula/FASTAS/GAPDH .
```

24) digite ls:

```
ls
```

### Editor de texto:

Vamos agora aprender a editar arquivos no servidor

25) Abrir o arquivo GAPDH no editor de texto vi:

```
vi GAPDH
```

1 - Para entrar no INSERT MODE e poder editar o texto digite " i "

2 - Troque o nome da sequência para >hsa

3 - Para salvar tecle **ESC** depois : (dois pontos), depois **x!** [enter]

26) Vamos ver o resultado:

```
more GAPDH
```

27) Vamos renomear o arquivo: :

```
mv GAPDH gapdh.hsa
```

28) Agora pegue a sequência de mioglobina:

1 - Entrar no navegador (pode ser o google chrome).

2 - Entrar no site:

<https://www.ncbi.nlm.nih.gov/nuccore/1049011000?report=fasta>

3 - Copiar a sequência fasta

29) Digite no terminal:

```
vi mioglobina
```

Cole o conteúdo do fasta. Botão da direita no terminal cola!!!!!!!  
Saia como antes ESC depois : depois x! [enter]

30) Troque o nome da sequência para ">myoglobin":

```
vi mioglobina
```

\*Não digite direto no terminal, antes digite "vi mioglobina"

31) Vamos ver o resultado:

```
more mioglobina
```

### O comando grep:

O comando grep é bastante utilizado para realizar buscas em textos/arquivos. A ideia é procurar um dado texto em uma string ou dentro de arquivos e mostrar as linhas de ocorrências:

32) Primeiro vamos ver o conteúdo do diretório /home/treinamento/blast\_aula/CDS:

```
ls /home/treinamento/blast_aula/CDS
```

33) Agora vamos ver o conteúdo do arquivo h.sapiens.nuc:

```
less /home/treinamento/blast_aula/CDS/h.sapiens.nuc
```

q (interrompe o comando)

34) Vamos identificar as ocorrências da palavra aldolase:

```
cat /home/treinamento/blast_aula/CDS/h.sapiens.nuc | grep aldolase
```

35) Vamos quantificar o número de sequências do arquivo fasta:

```
cat /home/treinamento/blast_aula/CDS/h.sapiens.nuc | grep ">" -c
```

## Referências:

- 1- ubuntu-br: <http://wiki.ubuntu-br.org/ComandosBasicos>
- 2- NCBI: <https://www.ncbi.nlm.nih.gov/>
- 3- Linuxbr: <http://br-linux.org/>

## Aula prática 2

- Sequence Quality Control
- Explorando dados de NGS

**Professor:** Jorge Estefano Santana de Souza, [jorge@imd.ufrn.br](mailto:jorge@imd.ufrn.br);

### Objetivos:

Utilizar as ferramentas básicas de análise de qualidade para obter um perfil inicial da qualidade do sequenciamento.

### Ferramentas:

- 1- Linux.
- 2- WebServer.
- 3- fastq\_screen:
- 4- fastqc
- 5- samstat
- 6- DynamicTrim.pl
- 7- trim\_galore
- 8- cutadapt

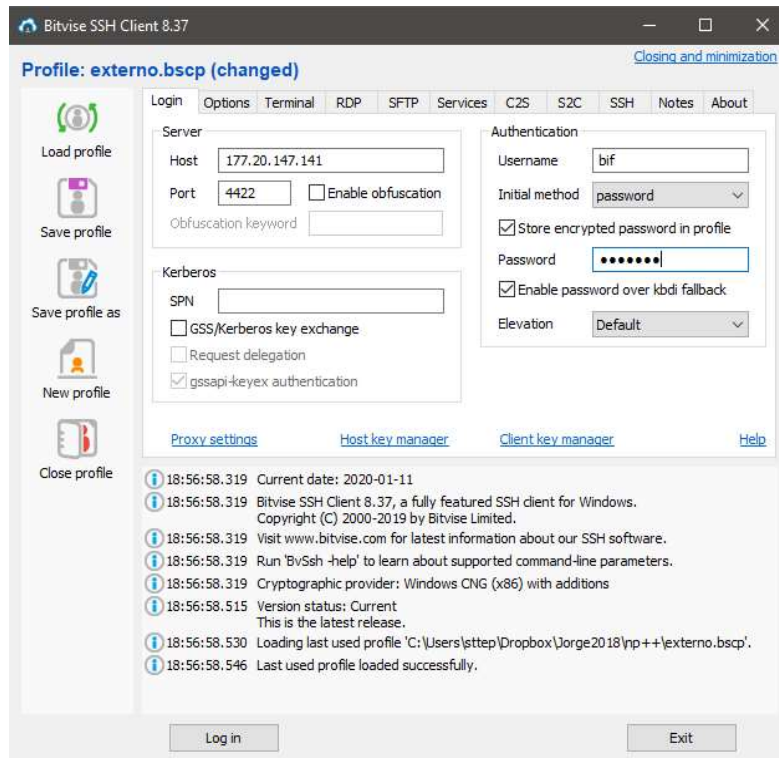
### Comandos Básicos:

Durante a execução dos tutoriais necessitaremos saber alguns comandos básicos do Linux. Procurar mais informação no site:

<http://wiki.ubuntu-br.org/ComandosBasicos>

### Login Servidor:

Inicialmente vamos fazer o login no servidor abrindo um terminal na máquina remota:



```
User:  bif
Host:  177.20.147.141
Porta: 4422
Senha: bif0003
```

### Dados brutos (raw data):

Durante a execução dos tutoriais necessitaremos de alguns dados iniciais, disponíveis no diretório:

```
/home/treinamento/NGS/
```

### Servidor WEB:

Como os trabalhos realizados no servidor são de difícil visualização, iremos necessitar de uma área web para facilitar nossa tarefa. Todos os arquivos copiados para o diretório .....

```
/data/home/bif/public_html/
```



.... estão disponíveis via navegador web em:

<http://www.bioinformatics-brazil.org/~bif/>



### Iniciando o Workflow:

1) Vamos começar pelo básico, certifique-se de que a pasta atual é:

```
/data/home/bif
```

Para isso digite o comando:

```
pwd
```

2) Caso não exista, crie um diretório contendo o seu nome digitando o comando:

```
mkdir SeuNome
```

3) Entre no diretório criado:

```
cd SeuNome
```

4) Crie um diretório chamado qual:

```
mkdir qual
```

5) Entre no diretório criado:

```
cd qual
```

6) Certifique-se de que a pasta atual é a correta:

```
pwd
```

O diretório atual deve ser: **/data/home/bif/SeuNome/qual**

7) Inicialmente, necessitaremos de arquivos no formato fastq, crie os links:

```
ln -s /home/treinamento/NGS/ERR844339.1.fastq .  
ln -s /home/treinamento/NGS/10_S5_R1_001.1.fastq .  
ln -s /home/treinamento/NGS/polipo.1.fastq .
```

8) Agora vamos ver o conteúdo de um arquivo fastq usando o comando:

```
less -S ERR844339.1.fastq
```

\*para sair digite a letra q

### Fastq\_screen:

9) Agora, vamos procurar por contaminações com o programa fastq\_screen usando o comando:

```
fastq_screen --nohits --subset 0 ERR844339.1.fastq --outdir .
```

\*o comando deve ser digitado em apenas uma linha

10) Podemos escolher os bancos de contaminantes, para tanto precisamos de um arquivo **.conf**. Copie o exemplo para o diretório corrente:

```
cp /home/treinamento/NGS/fastq_screen.conf .
```

11) Apague o resultado anterior:

```
rm ERR844339.1_*
```

12) Edite o arquivo `fastq_screen.conf` com o comando:

```
nano fastq_screen.conf
```

Esse comando abre o editor de texto nano mostrando o conteúdo do arquivo `fastq_screen.conf`. Dentro desse arquivo vamos substituir uma das primeiras linhas onde está escrito.

```
DATABASE c.elegans /home/databases/c_elegans/c_elegans BOWTIE2
```

Por:

```
#DATABASE c.elegans /home/databases/c_elegans/c_elegans BOWTIE2
```

Salve e saia do nano.

13) Agora, execute o comando:

```
fastq_screen --nohits --subset 0 --conf fastq_screen.conf  
ERR844339.1.fastq --outdir .
```

\*o comando deve ser digitado em apenas uma linha

14) Para melhor visualização do resultado vamos copiá-los para nossa área web:

Primeiro crie um diretório:

```
mkdir /data/home/bif/public_html/SeuNome
```

Depois copie:

```
cp ERR844339.1_* /home/bif/public_html/SeuNome
```

Agora entre no navegador web e visualize no endereço:

```
http://www.bioinformatics-brazil.org/~bif/SeuNome/
```

### Fastqutils:

15) Algumas estatísticas básicas podem ser obtidas com o programa fastqutils:

```
fastqutils stats ERR844339.1.fastq | more
```

```
fastqutils stats ERR844339.1_no_hits.fastq | more
```

### FastQC:

16) Para ter uma estatística mais ampla do resultado do sequenciamento utilizamos o programa FastQC:

```
fastqc ERR844339.1.fastq -o .
```

17) O resultado dele é um HTML, melhor visualizado na área web. Então, vamos copiar o arquivo para lá com o comando:

```
cp ERR844339.1_fastqc.* /home/bif/public_html/SeuNome/
```

18) Repita o processo para os demais arquivos **.fastq** e visualize as diferenças:

```
fastqc 10_S5_R1_001.1.fastq -o .  
  
fastqc polipo.1.fastq -o .  
  
cp *_fastqc.* /home/bif/public_html/SeuNome/
```

### **SAMstat:**

19) O programa **FastQC** é um programa pré-alinhamento, para avaliar as sequências alinhadas podemos utilizar o programa samstat:

Antes, vamos necessitar de um arquivo já alinhado (arquivo BAM). Para isso, crie o link:

```
ln -s /home/treinamento/NGS/polipo.bam .
```

Rode o samstat:

```
samstat polipo.bam
```

Não se esqueça de copiar o resultado para a área web:

```
cp *.samstat.* /home/bif/public_html/SeuNome/
```

\*veja o resultado em: <http://www.bioinformatics-brazil.org/~bif/SeuNome/>

### **Trim:**

Após o uso dos programas **FastQC** e **Samstat** talvez seja necessário aplicar algum processo de limpeza com o objetivo de melhorar os resultados finais e diminuir a taxa de erro de análises posteriores.

20) Podemos usar o **DynamicTrim** para trimar as pontas das sequências por qualidade de bases:

```
DynamicTrim.pl -h 20 ERR844339.1.fastq
```

21) Podemos usar o **cutadapt** para remover adaptadores ou sequências contaminantes conhecidas.

```
cutadapt -a TGGAATTCTCGG 10_S5_R1_001.1.fastq >  
10_S5_R1_001.1.ct.fastq
```

\*o comando deve ser digitado em apenas uma linha

22) O programa **trim\_galore** pode ser usado quando temos um sequenciamento Illumina e não sabemos os adaptadores usados no processo de sequenciamento:

```
trim_galore 10_S5_R1_001.1.fastq
```

Vale ressaltar que após cada processo de limpeza devemos refazer a análise de qualidade novamente, e assim verificar a sua efetiva melhora.

### Referências:

- 1- Lassmann et al. (2010) "SAMStat: monitoring biases in next generation sequencing data." Bioinformatics doi:10.1093/bioinformatics/btq614 [PMID: 21088025]
- 2- fastq\_screen: [https://www.bioinformatics.babraham.ac.uk/projects/fastq\\_screen](https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen)
- 3- fastqc: <https://www.bioinformatics.babraham.ac.uk/projects/fastqc>
- 4- trim\_galore: [https://www.bioinformatics.babraham.ac.uk/projects/trim\\_galore](https://www.bioinformatics.babraham.ac.uk/projects/trim_galore)
- 5- Cutadapt removes adapter sequences from high-throughput sequencing reads. MARTIN, Marcel. EMBnet.journal, [S.l.], v. 17, n. 1, p. pp. 10-12, may. 2011. ISSN 2226-6089. Available at: <<http://journal.embnet.org/index.php/embnetjournal/article/view/200>>. Date accessed: 08 Jul. 2017. doi:<http://dx.doi.org/10.14806/ej.17.1.200>.
- 6- samstat: <https://samstat.sourceforge.net>
- 7- DynamicTrim.pl: <https://github.com/hanice/SIBS/blob/master/Sequencing/QC/DynamicTrim.pl>

## Aula prática 3

- Chamada de variantes
- Explorando dados de NGS

**Professor:** Jorge Estefano Santana de Souza, [jorge@imd.ufrn.br](mailto:jorge@imd.ufrn.br);

### Objetivos:

Utilizar ferramentas básicas de chamada de variantes e identificar bases variantes em um sequenciamento de segunda geração.

### Ferramentas:

- 1- Linux.
- 2- WebServer.
- 3- bwa
- 4- samtools
- 5- mpileup
- 6- VarScan
- 7- SnpEff

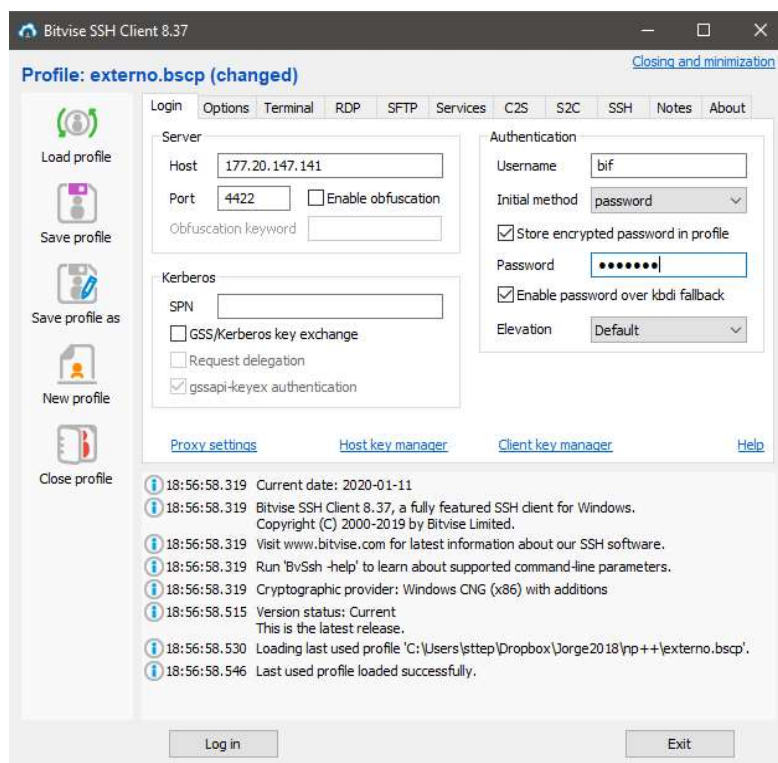
### Comandos Básicos:

Durante a execução dos tutoriais necessitaremos saber alguns comandos básicos do Linux. Mais informações no site:

<http://wiki.ubuntu-br.org/ComandosBasicos>

### Login Servidor:

Inicialmente vamos fazer o logon no servidor, abra um terminal na máquina remota:



```
User:  bif
Host:  177.20.147.141
Porta: 4422
Senha: bif0003
```

### Dados brutos (raw data):

Durante a execução dos tutoriais necessitaremos de alguns dados iniciais, disponíveis no diretório:

```
/home/treinamento/NGS/
```

### Servidor WEB:

Como os trabalhos realizados no servidor são de difícil visualização, iremos necessitar de uma área web para facilitar nossa tarefa. Todos os arquivos copiados para o diretório.....

```
/data/home/bif/public_html/
```



.... estarão disponíveis via navegador web em:

<http://www.bioinformatics-brazil.org/~bif/>



### Iniciando o Workflow:

1) Vamos começar pelo básico, certifique-se de que a pasta atual é:

```
/data/home/bif
```

Para isso, digite o comando:

```
pwd
```

2) Crie um diretório contendo o seu nome (se já não existe), digite o comando:

```
mkdir SeuNome
```

3) Entre no diretório criado:

```
cd SeuNome
```

4) Crie um diretório chamado bwa:

```
mkdir bwa
```

5) Entre no diretório criado:

```
cd bwa
```

6) Certifique-se de que a pasta atual é a correta:

```
pwd
```

O diretório atual deve ser: **/data/home/bif/SeuNome/bwa**

7) Crie links simbólicos para os arquivos:

```
ln -s /home/treinamento/NGS/hg19_chr8.1.fa .  
ln -s /home/treinamento/NGS/proband_R1.fq .  
ln -s /home/treinamento/NGS/proband_R2.fq .  
ln -s /home/treinamento/NGS/mother_R1.fq .  
ln -s /home/treinamento/NGS/mother_R2.fq .  
ln -s /home/treinamento/NGS/father_R1.fq .  
ln -s /home/treinamento/NGS/father_R2.fq .
```

8) Agora vamos ver o conteúdo de um arquivo fasta com o comando:

```
less -S hg19_chr8.1.fa
```

\*para sair digite a letra q



```
samtools faidx hg19_chr8.1.fa
```

11) Agora vamos rodar BWA utilizando os arquivos gerados até aqui:

```
bwa bwasw -t 4 hg19_chr8.1.fa \
proband_R1.fq \
proband_R2.fq -f proband.sam

bwa bwasw -t 4 hg19_chr8.1.fa \
mother_R1.fq \
mother_R2.fq -f mother.sam

bwa bwasw -t 4 hg19_chr8.1.fa \
father_R1.fq \
father_R2.fq -f father.sam
```

\*o comando deve ser digitado em apenas uma linha

### Utilizando o Samtools (analisando o alinhamento):

Agora que temos o arquivo SAM vamos convertê-lo para BAM utilizando o Samtools para manipulá-lo e extrair algumas estatísticas básicas.

\*ps. informação format .bam em: [http://genome.sph.umich.edu/wiki/SAM\\_Format](http://genome.sph.umich.edu/wiki/SAM_Format)

12) Convertendo de SAM para BAM:

```
samtools view -b -S proband.sam -o proband.bam
samtools view -b -S mother.sam -o mother.bam
samtools view -b -S father.sam -o father.bam
```

13) Visualizando um arquivo BAM:

```
samtools view proband.bam | less -S
```

14) Visualizando apenas as sequências não mapeadas:

```
samtools view -f 4 proband.bam | less -S
```

15) Visualizando apenas as sequências mapeadas:

```
samtools view -F 4 proband.bam | less -S
```

16) Quantificando as sequências não mapeadas:

```
samtools view -c -f 4 proband.bam
```

17) Quantificando as sequências com qualidade MAPQ superior a 42:

```
samtools view -c -q 42 proband.bam
```

**Atividade, responda:**

Quantas sequências foram mapeadas no genoma referência?

Quantas sequências foram mapeadas com qualidade superior a MAPQ 30?

Quantos pareamentos corretos existem?

**Em busca das variantes:**

Agora vamos tentar identificar as variantes genômicas. Para tanto temos que gerar o arquivo mpileup, mas antes temos que ordenar as sequências do arquivo BAM e remover a amplificação de PCR.

18) Ordenando as sequências do arquivo BAM:

```
samtools sort proband.bam > proband.sort.bam  
samtools sort mother.bam > mother.sort.bam  
samtools sort father.bam > father.sort.bam
```

## 19) Removendo as duplicações resultantes da amplificação de PCR:

```
samtools rmdup proband.sort.bam proband.rm.bam
samtools rmdup mother.sort.bam mother.rm.bam
samtools rmdup father.sort.bam father.rm.bam
```

## 20) Gerando o arquivo mpileup:

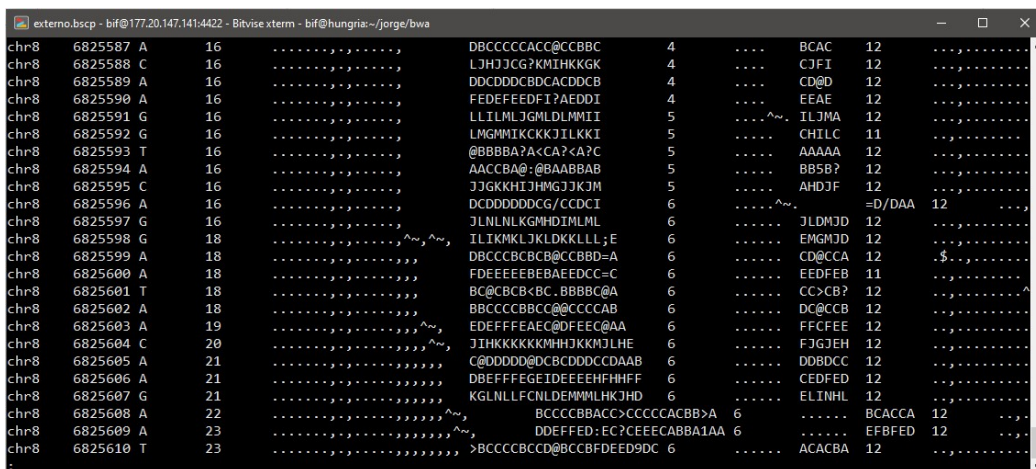
```
samtools mpileup -f hg19_chr8.1.fa proband.rm.bam
mother.rm.bam father.rm.bam > samples.mpileup
```

\*o comando deve ser digitado em apenas uma linha

## 21) Agora vamos ver o conteúdo do arquivo mpileup com o comando:

```
less -S samples.mpileup
```

\*para sair digite a letra q



```
chr8 6825587 A 16 ..... DBCCCCCACC@CCBBC 4 ..... BCAC 12 .....
chr8 6825588 C 16 ..... L3H7JCG?KMIHKKGK 4 ..... CJFI 12 .....
chr8 6825589 A 16 ..... DDCDDDCBDCACDDC 4 ..... CD@D 12 .....
chr8 6825590 A 16 ..... FEDEFEDETI?AEDDI 4 ..... EEAE 12 .....
chr8 6825591 G 16 ..... L1L1MLJGMLDLNMI 5 ..... ILJMA 12 .....
chr8 6825592 G 16 ..... LMGMIMKCKJILKKI 5 ..... CHILC 11 .....
chr8 6825593 T 16 ..... @BBBBBA?A<CA?<A?C 5 ..... AAAAA 12 .....
chr8 6825594 A 16 ..... AACCBAA@:@BAABBAB 5 ..... BB5B? 12 .....
chr8 6825595 C 16 ..... JJGKKH1JHMGJJKJM 5 ..... AHDJF 12 .....
chr8 6825596 A 16 ..... DCCCCDDDCG/CCDCI 6 ..... =D/DAA 12 .....
chr8 6825597 G 16 ..... J1NLNLKGMDI1ML 6 ..... J1DMJD 12 .....
chr8 6825598 G 18 ..... ILIKMKLJKLDKLL;E 6 ..... EMGMJD 12 .....
chr8 6825599 A 18 ..... DBCCCBBCB@CCBB0=A 6 ..... CD@CCA 12 .....
chr8 6825600 A 18 ..... FDEEEEBEBAEEDCC=C 6 ..... EEDFEB 11 .....
chr8 6825601 T 18 ..... BC@CBCK<BC. BB8BC@A 6 ..... CC>CB? 12 .....
chr8 6825602 A 18 ..... BBCCCBBC@@CCCCAB 6 ..... DC@CCB 12 .....
chr8 6825603 A 19 ..... EDEFFFEAC@DFEEC@AA 6 ..... FFCFEE 12 .....
chr8 6825604 C 20 ..... JIHKKKKKKMHJKMJLHE 6 ..... FJGJEH 12 .....
chr8 6825605 A 21 ..... C@DDDD@DCBCDDCCDAAB 6 ..... DDBDCC 12 .....
chr8 6825606 A 21 ..... DBEFFFEIDEEEFHFF 6 ..... CEDFED 12 .....
chr8 6825607 G 21 ..... KGLNLLFCNLDEMMMLHKJHD 6 ..... ELINHL 12 .....
chr8 6825608 A 22 ..... BCCCCBACC>CCCCACBB>A 6 ..... BCACCA 12 .....
chr8 6825609 A 23 ..... DDEFFED:EC?CEEECABBA1AA 6 ..... EBFED 12 .....
chr8 6825610 T 23 ..... >BCCCCBCCD@BCCBFDEED9DC 6 ..... ACACBA 12 .....
```

## 22) Agora, vamos fazer a chamada de variantes usando o programa VarScan:

```
varscan mpileup2snp samples.mpileup -output-vcf >
samples.vcf
```

\*o comando deve ser digitado em apenas uma linha

## Anotação das variantes:

O arquivo VCF (variant call format), contém todas as variantes (de base única, de inserção e de deleção) identificadas em uma ou mais amostras. No entanto, nessa versão inicial não estão anotadas todas as informações relevantes para extrair o significado biológico de cada variante. Para tanto, devemos executar o processo de anotação de variantes.

23) Agora fazer a anotação das variantes usando o programa SnpEff:

```
snpEff eff hg19 samples.vcf > samples.eff.vcf
```

\*o comando deve ser digitado em apenas uma linha

23) Vamos visualizar os arquivos e verificar as diferenças:

```
less -S samples.vcf  
  
less -S samples.eff.vcf
```

\*o comando deve ser digitado em apenas uma linha

Agora temos o arquivo que contém todas as variantes e as informações relevantes para extrair o significado biológico de cada variante.

Por fim tente isso:

```
grep -v "^#" samples.eff.vcf | cut -f 10,11,12 | grep  
-v "\./\." | sed "s/\:/\t/g" | cut -f 1,15,29 | more
```

## Referências:

- 1- Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics, 25:1754-60. [PMID: [19451168](#)]
- 2- Li H. and Durbin R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler Transform. Bioinformatics, Epub. [PMID: 20080505]
- 3- Li H.\*, Handsaker B.\*, Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9. [PMID: 19505943]
- 4- Li H A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011 Nov 1;27(21):2987-93. Epub 2011 Sep 8. [PMID: 21903627]

- 5- VarScan 1: Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, Weinstock GM, Wilson RK, & Ding L (2009). VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* (Oxford, England), 25 (17), 2283-5 PMID: 19542151
- 6- VarScan 2: Koboldt, D., Zhang, Q., Larson, D., Shen, D., McLellan, M., Lin, L., Miller, C., Mardis, E., Ding, L., & Wilson, R. (2012). VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing *Genome Research* DOI: 10.1101/gr.129684.111 URL: <http://varscan.sourceforge.net>
- 7- A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3.", Cingolani P, Platts A, Wang le L, Coon M, Nguyen T, Wang L, Land SJ, Lu X, Ruden DM. *Fly* (Austin). 2012 Apr-Jun;6(2):80-92. PMID: 22728672



## Aula prática 4

- RNAseq
- Explorando dados de NGS

**Professor:** Jorge Estefano Santana de Souza, [jorge@imd.ufrn.br](mailto:jorge@imd.ufrn.br);

### Objetivos:

Utilizar as ferramentas básicas de alinhamento e montagem de transcriptoma para identificar os genes diferencialmente expressos entre duas amostras.

### Ferramentas:

- 1- Linux.
- 2- WebServer.
- 3- tophat2
- 4- cufflinks
- 5- cuffmerge
- 6- cuffdiff
- 7- trimmomatic

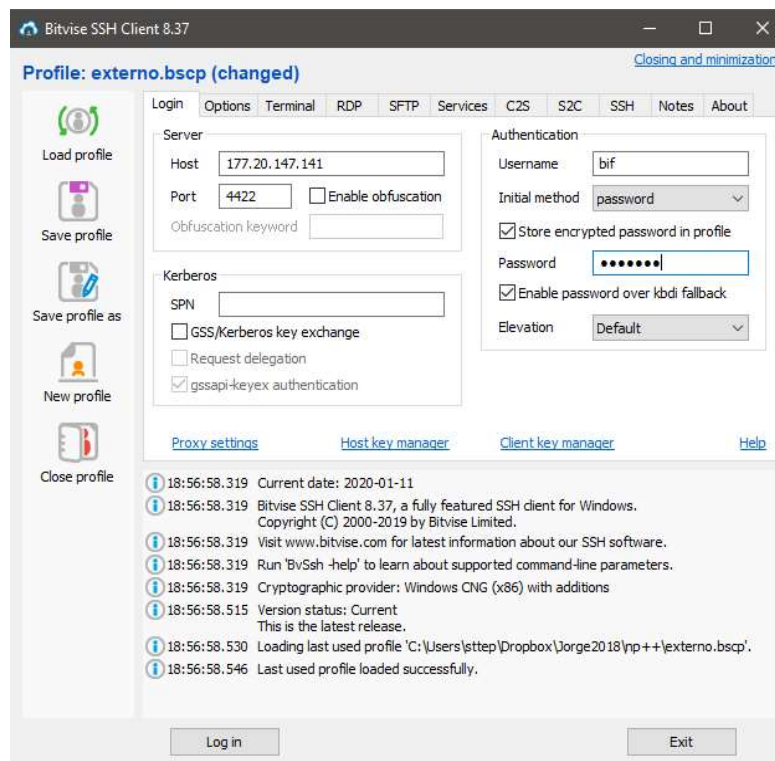
### Comandos Básicos:

Durante a execução dos tutoriais necessitaremos saber alguns comandos básicos do Linux. Mais informação no site:

<http://wiki.ubuntu-br.org/ComandosBasicos>

### Login Servidor:

Inicialmente vamos fazer o logon no servidor abrindo um terminal na máquina remota:



```
User:  bif
Host:  177.20.147.141
Porta: 4422
Senha: bif0003
```

### Dados brutos (raw data):

Durante a execução dos tutoriais necessitaremos de alguns dados iniciais, disponíveis no diretório:

```
/home/treinamento/NGS/
```

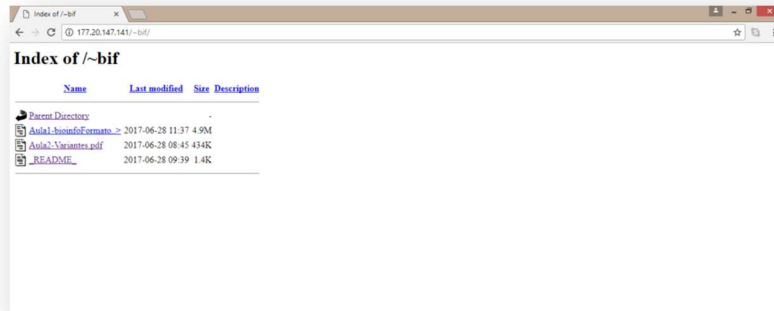
### Servidor WEB:

Como os trabalhos realizados no servidor são de difícil visualização, iremos necessitar de uma área web para facilitar nossa tarefa. Todos os arquivos copiados para o diretório abaixo....

```
/data/home/bif/public_html/
```

...estão disponíveis via navegador web em:

```
http://www.bioinformatics-brazil.org/~bif/
```



### Iniciando o Workflow:

1) Vamos começar pelo básico, certifique-se de que a pasta atual é:

```
/data/home/bif
```

Para isso, digite o comando:

```
pwd
```

2) Crie um diretório contendo o seu nome (se já não existe) digitando o comando:

```
mkdir SeuNome
```

3) Entre no diretório criado:

```
cd SeuNome
```

4) Crie um diretório chamado rna1:

```
mkdir rna1
```

5) Entre no diretório criado:

```
cd rna1
```

6) Certifique-se de que a pasta atual é a correta:

```
pwd
```

O diretório atual deve ser: /data/home/bif/SeuNome/rna1

7) Crie links simbólicos para os arquivos:

```
ln -s /home/treinamento/NGS/RNAseq/adrenal_1.fastq .  
ln -s /home/treinamento/NGS/RNAseq/adrenal_2.fastq .  
ln -s /home/treinamento/NGS/RNAseq/brain_1.fastq .  
ln -s /home/treinamento/NGS/RNAseq/brain_2.fastq .
```

8) Agora vamos ver o conteúdo de um arquivo fastq com o comando:

```
less -S adrenal_1.fastq
```

\*para sair digite a letra q

```

@ERR030881.107 HWI-BRUNOP16X_0001:2:1:13663:1096#0/1
ATCTTTTGTGGCTACAGTAAGTTCAATCTGAAGTCAAAACCAACCAATTT
+
5.544,444344555CC?CAEF@EEEEEEEEEEEEEEEEEEEEEEEEEEEE
@ERR030881.311 HWI-BRUNOP16X_0001:2:1:18330:1130#0/1
TCCATACATAGGCCTCGGGTGGGGGAGTCAGAAGCCCCCAGACCCTGTG
+
GFFFGFFBFCHHHHHHHHHHHIHEEE@@@=GHGHHHHHHHHHHHHHHHH
@ERR030881.1487 HWI-BRUNOP16X_0001:2:1:4144:1420#0/1
GTATAACGCTAGACACAGCGGAGCTCGGGATTGGCTAAACTCCCATAGTA
+
55*'+'&&5'55(''888:8FFFFFFFFF4/1;/4./++FFFFF=5:E#
@ERR030881.9549 HWI-BRUNOP16X_0001:2:1:1453:3458#0/1
AACGGATCCATTGTTTCGAGAACGTGATCGCCCTCATCTACCTAGCCTCA
+
D<@DDA@A:AAAAAAAAAAAAAAAAHHHHHHHHHHHHHHHHHHHHHH
@ERR030881.13497 HWI-BRUNOP16X_0001:2:1:16344:4145#0/1
GCTAATCCGACTTCTCGCCATCATCCTCCTGGTGGGTGTCACCATCGTGC
:

```

\*mais informação sobre o formato FASTQ em  
[https://en.wikipedia.org/wiki/FASTQ\\_format](https://en.wikipedia.org/wiki/FASTQ_format).

9) Faça o mesmo para os outros arquivos.

```

less -S adrenal_2.fastq

less -S brain_1.fastq

less -S brain_2.fastq

```

\*ps. para sair digite a letra q

10) Seria interessante saber o número de sequências totais nos arquivos. Para isso temos o comando `wc nome_do_arquivo` (word count):

```
wc adrenal_1.fastq
```

O problema aqui é que o comando conta o número de linhas totais. Mas podemos utilizar uma união com o comando `grep` (para procurar apenas o cabeçalho das reads). E só depois fazer a contagem.

```
grep '@ERR' adrenal_1.fastq | wc
```

## Filtragem:

11) Vamos analisar as sequências de entrada:

Use o comando `fastqc` no terminal para checar a qualidade do sequenciamento.

12) No próximo passo vamos filtrar os arquivos fastq. Essa etapa é importante para a diminuição dos erros gerados durante o sequenciamento. A ferramenta que iremos utilizar nessa etapa será o `trimmomatic`. O comando abaixo faz:

- Remoção de adaptadores;
- Remoção de bases do início com baixa qualidade ou Ns;
- Remoção de bases do fim com baixa qualidade ou Ns;
- Percorre o read com uma janela de 4, removendo quando a qualidade média por base é menor do que phd 15;
- Descarta reads com comprimento menor do que 20 bases.

Comando 1 (link para adaptadores):

```
ln -s /data/home/root/Trimmomatic-0.36/adapters/TruSeq3-PE-2.fa .
```

Comando 2 (trim):

```
trimmomatic PE -threads 1 \
    adrenal_1.fastq adrenal_2.fastq \
    adrenal_1_paired.fastq.gz adrenal_1_unpaired.fastq.gz \
    adrenal_2_paired.fastq.gz adrenal_2_unpaired.fastq.gz \
    ILLUMINACLIP:TruSeq3-PE-2.fa:2:30:10 \
    LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:20
```

\*o comando deve ser digitado em apenas uma linha

Repita o passo anterior com os dados de cérebro:

```
trimmomatic PE -threads 1 \
    brain_1.fastq brain_2.fastq \
    brain_1_paired.fastq.gz brain_1_unpaired.fastq.gz \
    brain_2_paired.fastq.gz brain_2_unpaired.fastq.gz \
    ILLUMINACLIP:TruSeq3-PE-2.fa:2:30:10 \
    LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:20
```

\*o comando deve ser digitado em apenas uma linha

13) Descompacte os arquivos que foram considerados filtrados e que mantiveram os pares após a filtragem:

```
gzip -d *_paired.fastq.gz
```

## Mapeamento:

14) Para realizar o mapeamento, primeiro temos que criar links simbólicos do genoma de referência e de nosso arquivo GTF (lembre-se de estar no diretório: /data/home/bif/SeuNome/rna1):

```
ln -s /home/databases/hg19/ ref
```

```
ln -s /home/treinamento/NGS/RNAseq/gene19_annotation.gtf .
```

15) Agora vamos rodar TopHat2 utilizando os arquivos gerados até aqui:

```
tophat -p 2 -G gene19_annotation.gtf \  
-o thout_adrenal \  
ref/hg19 \  
adrenal_1_paired.fastq \  
adrenal_2_paired.fastq
```

\*o comando deve ser digitado em apenas uma linha

Esse passo pode demorar. Uma alternativa é copiar os arquivos prontos de: /home/treinamento/NGS/

```
cp -r /home/treinamento/NGS/biome/rna1/thout_adrenal/ .
```

Faremos o mesmo com a amostra de cérebro:

```
tophat -p 2 -G gene19_annotation.gtf \  
-o thout_brain \  
ref/hg19 \  
brain_1_paired.fastq \  
brain_2_paired.fastq
```

\*o comando deve ser digitado em apenas uma linha

Ou copiar os arquivos prontos de: /home/treinamento/NGS/

```
cp -r /home/treinamento/NGS/biome/rna1/thout_brain/ .
```

### Montagem:

16) A montagem dos transcritos pode ser feita com a ferramenta Cufflinks:

```
cufflinks -p 4 -o clout_adrenal thout_adrenal/accepted_hits.bam
```

Repita o passo com a amostra de cérebro:

```
Cufflinks -p 4 -o clout_brain thout_brain/accepted_hits.bam
```

### Expressão diferencial:

17) Quais genes estão diferencialmente expressos? Para responder a pergunta, primeiro vamos criar os arquivos de input para o cuffdiff:

```
samtools view -h thout_adrenal/accepted_hits.bam > adrenal.sam  
samtools view -h thout_brain/accepted_hits.bam > brain.sam
```

18) Agora vamos rodar o Cuffdiff:

```
cuffdiff -o diff_out gene19_annotation.gtf adrenal.sam brain.sam
```

\*o comando deve ser digitado em apenas uma linha

### Olhando o Resultado:

19) Dentro da pasta diff\_out encontramos vários resultados interessantes. Com ls podemos checar os arquivos gerados.

```
ls diff_out
```

Vamos manter o foco no arquivo gene\_exp.diff .



```
more diff_out/gene_exp.diff
```

Agora vamos selecionar apenas aqueles genes dados como diferencialmente expressos pelos testes estatísticos do cuffdiff.

```
grep 'yes$' diff_out/gene_exp.diff
```

### Referências:

- 1- Differential gene and transcript expression analysis of RNAseq Experiments with TopHat and Cufflinks. Trapnell C 1 , Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L.
- 2- Trimmomatic: A flexible trimmer for Illumina Sequence Data. Anthony M. Bolger, Marc Lohse and Bjoern Usadel
- 3- Simple Combinations of LineageDetermining Transcription Factors Prime cisRegulatory Elements Required for Macrophage and B Cell Identities. Heinz S, Benner C, Spann N, Bertolino E et al.
- 4- Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Cole Trapnell, Brian Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Jeltje van Baren, Steven Salzberg, Barbara Wold, Lior Pachter. Nature Biotechnology, 2010.

## Aula prática 5

- RNAseqII
- Explorando dados de NGS

**Professor:** Jorge Estefano Santana de Souza, [jorge@imd.ufrn.br](mailto:jorge@imd.ufrn.br);

### Objetivo:

Utilizar as ferramentas básicas de RNAseq para obter o padrão de expressão dos mirRNAs de uma amostra.

### Ferramentas:

- 1- Linux.
- 2- WebServer.
- 3- cutadapt
- 4- mapper
- 5- miRDeep2

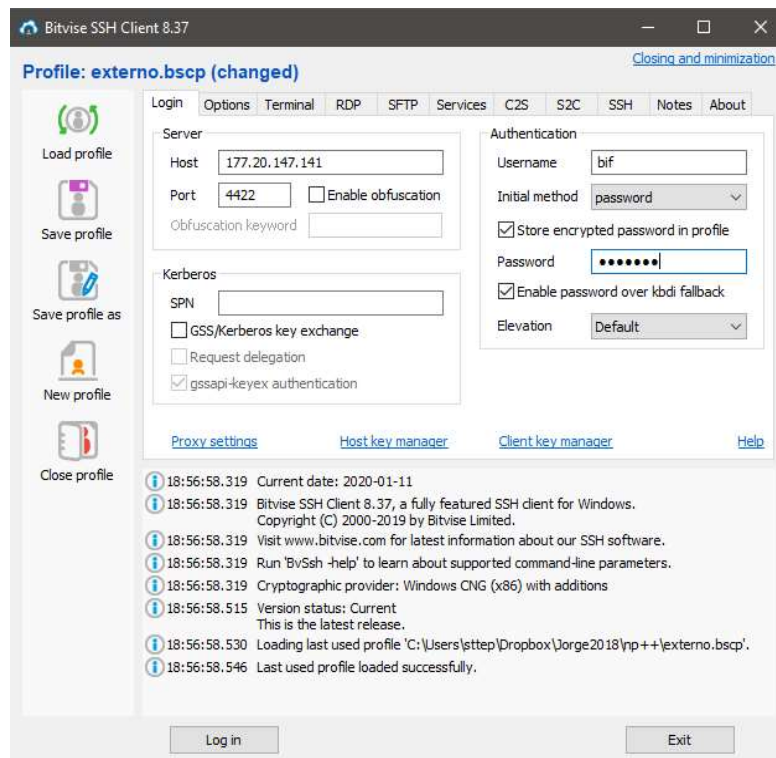
### Comandos Básicos:

Durante a execução dos tutoriais necessitaremos saber alguns comandos básicos do Linux. Mais informações no site:

<http://wiki.ubuntu-br.org/ComandosBasicos>

### Login Servidor:

Inicialmente vamos fazer o logon no servidor abrindo um terminal na máquina remota:



```
User:  bif
Host:  177.20.147.141
Porta: 4422
Senha: bif0003
```

### Dados brutos (raw data):

Durante a execução dos tutoriais necessitaremos de alguns dados iniciais disponíveis no diretório:

```
/home/treinamento/NGS/
```

### Servidor WEB:

Como os trabalhos realizados no servidor são de difícil visualização, iremos necessitar de uma área web para facilitar nossa tarefa. Todos os arquivos copiados para o diretório abaixo...

```
/data/home/bif/public_html/
```

....estarão disponíveis via navegador web em:

```
http://www.bioinformatics-brazil.org/~bif/
```



### Iniciando o Workflow:

1) Vamos começar pelo básico, certifique-se de que a pasta atual é:

```
/home/bif
```

Para isso digite o comando:

```
pwd
```

2) Crie um diretório contendo o seu nome (se já não existir) digitando o comando:

```
mkdir SeuNome
```

3) Entre no diretório criado:

```
cd SeuNome
```

4) Crie um diretório chamado rna2:

```
mkdir rna2
```

5) Entre no diretório criado:

```
cd rna2
```

6) Certifique-se de que a pasta atual é a correta:

```
pwd
```

O diretório atual deve ser: **/data/home/bif/SeuNome/rna2**

7) Crie links simbólicos para os arquivos:

```
ln -s /home/treinamento/NGS/RNAseq/sample_data/SRR326279_R1.fastq .  
ln -s /home/treinamento/NGS/RNAseq/sample_data/SRR326280_R1.fastq .
```

\*esses são os sequenciamentos de nossas amostras.

### Arquivos de Referência:

8) Agora vamos necessitar de:

- Arquivo fasta com o genoma de referência;
- Arquivos de index do genoma de referência;
- Arquivo fasta com os miRNAs referência para a espécie (utilizaremos o miRBase);
- Arquivo fasta com os miRNAs maduros para a espécie (utilizaremos o miRBase);
- Arquivo fasta de predição dos loops das sequências dos miRNAs para a espécie, os hairpins (utilizaremos o miRBase);

Esses arquivos já foram baixados. Crie um link para eles:

```
ln -s /home/treinamento/NGS/RNAseq/small_ref/ .
```

## Preparando o dado inicial:

9) Antes de executar o miRDeep2, os dados devem ser pré-processados para remover adaptadores. Isso pode ser feito usando o cutadapt:

```
cutadapt -b AATCTCGTATGCCGTCTTCTGCTTGC -O 3 -m 17 \
-f fastq SRR326279_R1.fastq > SRR326279_R1.ct.fastq
```

\*o comando deve ser digitado em apenas uma linha

```
cutadapt -b AATCTCGTATGCCGTCTTCTGCTTGC -O 3 -m 17 \
-f fastq SRR326280_R1.fastq > SRR326280_R1.ct.fastq
```

## Mapeamento:

10) Usaremos o script mapper.pl para processar as leituras e mapeá-las contra o genoma de referência:

```
mapper.pl SRR326279_R1.ct.fastq -e \
-p small_ref/hg19_chr1 -s SRR326279.pr.fa \
-t SRR326279.mr.arf -h -m -i -j
```

\*o comando deve ser digitado em apenas uma linha

```
mapper.pl SRR326280_R1.ct.fastq -e \
-p small_ref/hg19_chr1 -s SRR326280.pr.fa \
-t SRR326280.mr.arf -h -m -i -j
```

\*o comando deve ser digitado em apenas uma linha

Identificação de miRNAs conhecidos e novos nos dados de sequenciamento:

```
miRDeep2.pl SRR326279.pr.fa small_ref/hg19_chr1.fa \
SRR326279.mr.arf small_ref/mature.hsa.dna.fa \
none small_ref/hairpin.hsa.dna.fa \
-t Human 2> report.log
```

\*o comando deve ser digitado em apenas uma linha

```
miRDeep2.pl SRR326280.pr.fa small_ref/hg19_chr1.fa \
SRR326280.mr.arf small_ref/mature.hsa.dna.fa \
none small_ref/hairpin.hsa.dna.fa \
-t Human 2> report.log
```

\*o comando deve ser digitado em apenas uma linha

**Esses dois últimos comandos irão demorar muito, muito mesmo, vamos pegar os resultados prontos em:**

```
ls /home/treinamento/NGS/biome/rna2/
```

Agora vamos olhar os resultados.

### Referências:

- 1- Simple Combinations of LineageDetermining Transcription Factors Prime cisRegulatory Elements Required for Macrophage and B Cell Identities. Heinz S, Benner C, Spann N, Bertolino E et al.
- 2- Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Cole Trapnell, Brian Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Jeltje van Baren, Steven Salzberg, Barbara Wold, LiorPachter. NatureBiotechnology, 2010
- 3- Cutadapt removes adapter sequences from high-throughput sequencing reads. MARTIN, Marcel. EMBnet.journal, [S.I.], v. 17, n. 1, p. pp. 10-12, may. 2011. ISSN 2226-6089. Available at:<<http://journal.embnet.org/index.php/embnetjournal/article/view/200>>. Date accessed: 08 Jul. 2017. doi:<http://dx.doi.org/10.14806/ej.17.1.200>.
- 4- Friedländer, M.R., Chen, W., Adamidi, C., Maaskola, J., Einspanier, R., Knespel, S., Rajewsky, N. 'Discovering microRNAs from deep sequencing data using miRDeep', Nature Biotechnology, 26, 407-415 (2008).