

# Aula prática: RNAseqI. Explorando dados de NGS.

06/07/2017

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

**Monitores:** Danilo Lopes Martins, [danilolmartins@gmail.com](mailto:danilolmartins@gmail.com);  
Luan Pereira, [luanpereira00@outlook.com](mailto:luanpereira00@outlook.com).

## Objetivos:

Utilizar as ferramentas básicas de alinhamento e montagem de transcriptoma para obter 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. Podem procurar mais informação no site:

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

## Login Servidor:

Inicialmente vamos fazer o login no servidor, abra um terminal no linux e digite:

```
ssh -p 4422 bif@10.7.5.38
```

Irá pedir uma senha, digite:

```
bif0003
```

\*ps. não aparece a digitação, o teclado não quebrou não!

## Regras para login no servidor:

Interno à UFRN:

```
ssh -p 4422 bif@10.7.5.38
```

Senha: bif0003

Externo à UFRN:

```
ssh -p 4422 bif@177.20.147.141
```

Senha: bif0003

## Dados brutos (raw data):

Durante a execução dos tutoriais necessitaremos de alguns dados iniciais, em via de regra estarão disponíveis no diretório:

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

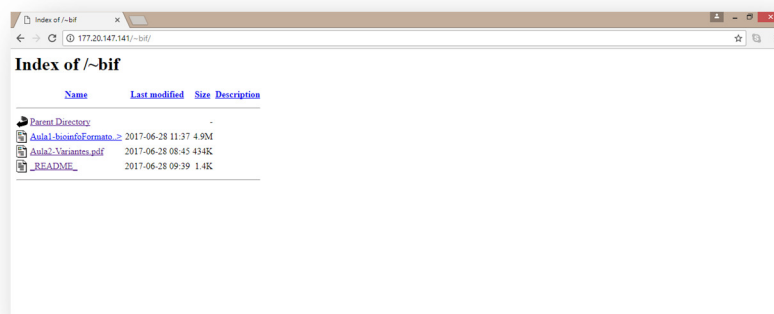
## Servido WEB:

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:

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

Estarão disponíveis via navegador web em:

```
http://177.20.147.141/~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, digite 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: /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 como um arquivo fastq é. comando:

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

\*ps. para sair digite a letra q



```
Bitwise xterm - externo.bsccp - bif@177.20.147.141:4422 - bif@zurique:...
@ERR030881.107 HWI-BRUNOP16X_0001:2:1:13663:1096#0/1
ATCTTTTGTGGCTACAGTAAGTTCAATCTGAAGTCAAACCAACCAATTT
+
5.544,444344555CC?CAEF@EEEEEEEEEEEEEEEEEEEEEEEEEEEE
@ERR030881.311 HWI-BRUNOP16X_0001:2:1:18330:1130#0/1
TCCATACATAGGCCTCGGGTGGGGGAGTCAGAAGCCCCAGACCCTGTG
+
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:AHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
@ERR030881.13497 HWI-BRUNOP16X_0001:2:1:16344:4145#0/1
GCTAATCCGACTTCTCGCCATCATCCTCCTGGTGGGTGTACCATCGTGC
:
```

\*ps. mais informação do 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 conseguir saber o numero de sequências totais nos arquivos. Temos algo para isso. 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), Vamos lá!

```
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 /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
```

\*ps. o comando deve ser digitado em apenas uma linha

Repita o passo anterior com os dados de cerebro:

```
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
```

\*ps. 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: /home/bif/SeuNome/rna1):

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

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

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

```
tophat -p 1 -G gene19_annotation.gtf
      -o thout_adrenal
      hg19/hg19
      adrenal_1_paired.fastq
      adrenal_2_paired.fastq
```

\*ps. o comando deve ser digitado em apenas uma linha

Esse passo pode demorar, uma alternativa seria copiar os arquivos prontos de: /home/treinamento/NGS/RNAseq/

Faremos o mesmo com a amostra de cérebro:

```
tophat -p 1 -G gene19_annotation.gtf  
-o thout_brain  
hg19/hg19  
brain_1_paired.fastq  
brain_2_paired.fastq
```

\*ps. o comando deve ser digitado em apenas uma linha

Esse passo pode demorar, uma alternativa seria copiar os arquivos prontos de: /home/treinamento/NGS/RNAseq/

### 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 um arquivo de anotação de referência para o cuffmerge:

```
nano assemblies.txt
```

nano é um editor de texto, e esse comando abre o editor criando o arquivo assemblies.txt. Dentro desse arquivo vamos escrever:

```
./clout_adrenal/transcripts.gtf  
./clout_brain/transcripts.gtf
```

(para sair pressionar Ctrl + x) Salvar pressionando Y

18) Vamos rodar o cuffmerge:

```
cuffmerge -g gene19_annotation.gtf -s hg19/hg19.fa  
-p 1 assemblies.txt
```

\*ps. o comando deve ser digitado em apenas uma linha

19) Agora vamos rodar o Cuffdiff:

```
cuffdiff -o diff_out  
-b hg19/hg19.fa  
-p 1  
-L A,B  
-u merged_asm/merged.gtf  
./thout_adrenal/accepted_hits.bam  
./thout_brain/accepted_hits.bam
```

\*ps. o comando deve ser digitado em apenas uma linha

## Olhando o Resultado:

14) Dentro da pasta diff\_out encontramos vários resultados interessantes. Com ls podemos checar alguns desses 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