

Tutorial Bioinformatica

Marcel Ferreira - Bolsista/CAPES

2023-11-30

Table of contents

Sobre esse curso	4
Realização	4
Apoio	4
Autores	4
Colaboradores	4
Introdução	5
Primeiros passos	6
Configurações de sistema	6
Softwares necessários	6
Usuários Windows	6
No Ubuntu	6
Instalação	7
Dados utilizados	8
Dia 1 - Sequenciamento de DNA	10
Arquivos	10
Métricas	10
Atividades	10
Dia 2 - Alinhamento de sequências de DNA	12
Arquivos	12
Atividades	12
Dia 3 - Genotipagem	14
Arquivos	14
Atividades	14
Dia 4 - Análise de sequenciamento Oxford Nanopore	16
Atividades	16
Dia 5 - Genotipagem de STRs a partir de dados de NGS	18
Referências	19

ANEXO: Comandos Básicos do Ubuntu via WSL	20
Navegação e Diretórios	20
Listar Conteúdo do Diretório	20
Mudar de Diretório	20
Diretório Atual	20
Criar Diretório	20
Manipulação de Arquivos	20
Copiar Arquivo	20
Mover/Renomear Arquivo	20
Remover Arquivo	21
Visualização de Conteúdo	21
Visualizar Conteúdo do Arquivo	21
Visualizar Conteúdo do Arquivo (página por página)	21
Pacotes e Atualizações	21
Atualizar Lista de Pacotes	21
Atualizar Pacotes Instalados	21
Instalar Novo Pacote	21
Gerenciamento de Usuários	21
Adicionar Usuário ao Grupo	21
Mudar Senha do Usuário	22
Processos	22
Listar Processos	22
Matar um Processo por ID	22
Monitorar Recursos do Sistema (htop)	22
Medir Tempo de Execução de um Comando (time)	22
Rede	22
Verificar Configurações de Rede	22
Testar Conexão com um Endereço IP	22
Entrada e Saída Padrão (stdin/stdout)	23
Outros Comandos Úteis	23
Ajuda sobre um Comando	23
Sair do Terminal	23

Sobre esse curso

Realização



Figure 1: Realização UnB, UNESP e USP.

Apoio



Figure 2: CAPES-PROCAD Edital n° 16/2020

Autores

Celso Teixeira Mendes Junior
Erick da Cruz Castelli
Marcel Rodrigues Ferreira
Tamara Soledad Frontanilla Recalde

Colaboradores

Introdução

Bem-vindos ao Workshop de Bioinformática Aplicada à Genética Forense: Análise de Dados de Sequenciamento de Segunda e Terceira Geração. Este curso abrangente foi projetado para fornecer a vocês uma imersão prática nas técnicas de análise de dados genômicos, com foco especial na aplicação forense.

A genética forense tornou-se uma ferramenta essencial na resolução de casos criminais, identificação de indivíduos e estabelecimento de relações familiares. Neste workshop de cinco dias, exploraremos os fundamentos e as aplicações práticas do sequenciamento de DNA, abordando desde os conceitos básicos até as técnicas avançadas de genotipagem de STRs (*Short Tandem Repeats*) a partir de dados de *Next-Generation Sequencing* (NGS).

Dia 1 - Sequenciamento de DNA: Iniciaremos nossa jornada explorando os princípios fundamentais do sequenciamento de DNA de segunda e terceira geração. Compreenderemos as tecnologias por trás desses métodos e sua importância na geração de dados genômicos de alta qualidade.

Dia 2 - Alinhamento de Sequências de DNA: No segundo dia, mergulharemos na etapa crucial de alinhamento de sequências de DNA. A precisão dessa fase é vital para extrair informações significativas dos dados brutos e identificar variações genéticas relevantes.

Dia 3 - Identificação de Variantes: Aprofundando-nos ainda mais, dedicaremos o terceiro dia à identificação de variantes genéticas. Exploraremos ferramentas e estratégias para detectar mutações, SNPs (*Single Nucleotide Polymorphisms*) e outras alterações que desempenham um papel crucial na individualidade genômica.

Dia 4 - Análise de Sequenciamento Oxford Nanopore: No quarto dia, abordaremos uma tecnologia revolucionária: o sequenciamento Oxford Nanopore. Compreenderemos suas vantagens, desafios e exploraremos casos de uso específicos na genética forense.

Dia 5 - Genotipagem de STRs a partir de dados de NGS: Encerraremos o workshop com uma exploração prática da genotipagem de STRs, uma ferramenta valiosa para estabelecer perfis genéticos únicos. Aprenderemos a interpretar e analisar esses dados, fornecendo insights fundamentais para investigações forenses.

Ao longo desta semana, vocês serão desafiados a aplicar os conhecimentos adquiridos em exercícios práticos e estudos de caso, preparando-os para enfrentar os desafios reais da genética forense na era da bioinformática avançada. Esteja preparado para uma jornada intensiva de aprendizado e descoberta!

Primeiros passos


Configurações de sistema

Antes de iniciarmos o tutorial, é imperativo garantir que o sistema atenda às configurações mínimas para uma experiência estável. Utilizaremos sistema Linux. Recomenda-se que a máquina disponha de, no mínimo, 30 GB de armazenamento e 8 GB de memória RAM. No entanto, para uma performance ideal e considerando o potencial de expansão das aplicações, encorajamos a utilização de um sistema com mais de 50 GB de armazenamento e, no mínimo, 16 GB de memória RAM. Essas configurações mais robustas assegurarão não apenas a instalação suave do software, mas também a capacidade de executar múltiplas aplicações de forma eficiente, proporcionando uma experiência mais fluida e responsiva ao usuário.

Softwares necessários

Usuários Windows

- WSL (Windows Subsystem for Linux)
- [IGV](#) (Robinson et al. 2011)
- [FASTQC](#)
- [notepad++](#)

 Tutorial para instalar o WSL

Siga o tutorial da microsoft para instalar o WSL.

<https://learn.microsoft.com/pt-br/windows/wsl/install>

No Ubuntu

- [IGV](#) (Robinson et al. 2011)
- [FASTQC](#)
- [Trimmomatic](#) (Bolger, Lohse, and Usadel 2014)
- [bwa](#) (Li 2013)
- [minimap2](#) (Li 2018, 2021)

- [samtools](#) (Danecek et al. 2021)
- [freebayes](#) (Garrison and Marth 2012)
- [gatk](#)
- [vcftools](#) (Danecek et al. 2011)
- [bcftools](#) (Danecek et al. 2021)
- [WhatsHap](#) (Martin et al. 2016)
- [NanoPlot](#) (De Coster and Rademakers 2023)
- [chopper](#) (De Coster and Rademakers 2023)
- [cramino](#) (De Coster and Rademakers 2023)

Opcionais

- [gzip](#)
- [HTSLib](#)

Instalação

Usuários windows

Usuários windows precisam instalar o Subsistema Windows para Linux (WSL). Os softwares FASTQC e IGV precisam ser instalados no windows e **não no WSL**.

Ao terminar a instalação do WSL e de configurar seu usuário no linux utilize os seguintes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Estes comandos irão garantir que o seu sistema esteja atualizado.

Sobre o comando `sudo`

O comando `sudo` permite ao usuário executar comandos com permissão superior. Para isso você precisará da sua **senha** (ou do administrador)!

Para instalar softwares utilize o comando `apt install` da seguinte forma:

```
sudo apt install [SOFTWARE]
```

Dados utilizados

Baixe os dados que serão utilizados neste workshop via [Google Drive](#);

! Utilize o email correto

Para ter acesso aos dados utilize o email que foi fornecido durante a inscrição no evento.
Em caso de erro entre em contato com a organização.
Os dados totalizam ~13 GB, se atente para isso.

Os dados estão contidos nesta estrutura de pastas descritas a baixo:

WorkshopDados/

```
--genome/
|  |--chr17.fas1
|  °--hg38.fa
|--fast5/
|  °--*arquivos.fast52
|--guppy_installer/
|  °--ont-guppy-cpu_6.5.7_linux64.tar.gz3
|--LongReadsFastq/
|  |--HG00096.hg38.fastq
|  °--HG00099.hg38.fastq
°--ShortReadsFastq/
|  |--HG00096_r1.fastq
|  |--HG00096_r2.fastq
|  |--HG00097_r1.fastq
|  |--HG00097_r2.fastq
|  |--HG00099_r1.fastq
|  |--HG00099_r2.fastq
|  |--HG00100_r1.fastq
|  |--HG00100_r2.fastq
|  |--NA18486_r1.fastq
|  |--NA18486_r2.fastq
|  |--NA18487_r1.fastq
|  |--NA18487_r2.fastq
|  |--NA18488_r1.fastq
|  |--NA18488_r2.fastq
|  |--NA19648_r1.fastq
|  |--NA19648_r2.fastq
```

¹Genoma do cromossomo 17 obtido em https://timkahlke.github.io/LongRead_tutorials/

²Arquivos **fast5** obtidos em https://timkahlke.github.io/LongRead_tutorials/

³Instalador obtido em <https://community.nanoporetech.com/>


```
|--NA19649_r1.fastq  
|--NA19649_r2.fastq  
|--NA19651_r1.fastq  
°--NA19651_r2.fastq
```

Dicas

Utilize o comando `htop` para monitorar o consumo de memória em seu computador durante todas as atividades deste tutorial.

Utilize os argumento `-h` ou `--help` para visualizar ajuda sobre o uso dos softwares via terminal.

SOFTWARE -h

Adicionar o comando `time` antes de rodar os códigos, para saber o tempo que demorou, irá te ajudar a se programar para atividades futuras.

`time [COMANDOS...]`

Dia 1 - Sequenciamento de DNA

! Importante

Verifique se o **FASTQC** esta instalado.

Arquivos

Serão utilizados os arquivos contidos na pasta **WorkshopDados/shortReads/**.

Métricas

$$Read\ Accuracy = \frac{N_{match}}{N_{match} + N_{mis} + N_{del} + N_{ins}} \quad (0.1)$$

$$Mis/Ins/Del = \frac{N_{mis/ins/del}}{N_{match} + N_{mis} + N_{del} + N_{ins}} \quad (0.2)$$

$$P = 10^{\frac{-Q_{score}}{10}} \quad (0.3)$$

$$Read\ Q_{score} = -10 \log_{10} \left[\frac{1}{N} \sum 10^{\frac{-q_1}{10}} \right] \quad (0.4)$$

Atividades

O controle de qualidade (QC) dos dados é uma etapa crítica na análise de sequenciamento de nova geração (NGS) para garantir a confiabilidade dos resultados. Abaixo estão as etapas típicas do controle de qualidade:

1. Análise Inicial com **FASTQC**:

- Execute o **FASTQC** nas suas leituras brutas para avaliar a qualidade geral. Isso inclui gráficos e estatísticas que indicam a distribuição da qualidade das bases ao longo das reads, a presença de adaptadores, a presença de sequências overrepresented, entre outros.

2. Identificação de Adaptação (Adapter) e Trimagem:

- Com base nos resultados do **FASTQC**, identifique a presença de adaptadores e sequências indesejadas nas extremidades das reads. Utilize o **Trimmomatic** para remover essas sequências, garantindo que apenas dados de alta qualidade sejam mantidos.
-

3. Remoção de Leituras de Baixa Qualidade:

- Algumas leituras podem conter regiões de baixa qualidade. Considere a remoção dessas leituras ou a trimagem de regiões específicas usando ferramentas adequadas, dependendo da natureza do problema.

4. Filtragem de Leituras Curtas ou Longas:

- Dependendo do seu experimento, você pode querer filtrar leituras muito curtas ou muito longas que possam representar artefatos ou problemas experimentais.

5. Avaliação de Qualidade Pós-Trimagem:

- Após a trimagem e filtragem, execute novamente o **FASTQC** para avaliar como essas etapas afetaram a qualidade dos dados. Isso ajudará a garantir que você atingiu os padrões de qualidade desejados.

6. Relatório Final de Controle de Qualidade:

- Compile todos os resultados de QC em um relatório final que destaque os principais aspectos da qualidade dos dados. Isso é útil para comunicação interna, bem como para garantir a transparência na publicação de resultados.

Dia 2 - Alinhamento de sequências de DNA

Arquivos

Os arquivos utilizados para estas análises serão os `.fastq` analisados no primeiro dia.

⚠ Arquivos `.fastq`

Preste atenção para o caminho da pasta aonde estão os `.fastq`.

Atividades

1. Indexação do Genoma de Referência:

- Antes de realizar o alinhamento, é necessário indexar o genoma de referência usando o comando `bwa index`. Isso cria arquivos que aceleram o processo de alinhamento.

```
bwa index reference_genome.fa
```

⚠ Indexação do genoma

O processo criar o índice do genoma via `bwa index` demora bastante tempo para ser realizado. Mesmo em máquinas com grandes capacidades de memória. Devido a isso colocamos na pasta **WorkshopDados/genome/** os arquivos resultantes desta etapa, que são os arquivos com extensão `.amb`, `.ann`, `.bwt`, `.fai`, `.pac` e `.sa`. Caso deseje testar em um genoma menor, utilize o `chr17.fas`.

2. Alinhamento de Sequências:

- Use o `bwa mem` para alinhar suas sequências de DNA ao genoma de referência.
- ```
bwa mem -R "@RG\tID:{SAMPLE}\tSM:{SAMPLE}" reference_genome.fa {SAMPLE}_r1.fq {SAMPLE}_r2.fq > alinhamento.sam
```

Substitua `reference_genome.fa`, `{SAMPLE}_r1.fq` e `{SAMPLE}_r2.fq` pelos nomes dos arquivos correspondentes.

### 💡 Automatizando o processo (Opcional)

Caso tenha experiência em programação, você pode utilizar um `loop`, como `for` ou `while`, para rodar todas as amostras ao mesmo tempo. Você pode utilizar o próprio `bash` (ubuntu) ou sua linguagem de programação favorita, como `python`, `perl`, `R`, etc.

Você poderá incluir as etapas seguintes no mesmo `loop`.

### 3. Converter Formato SAM para BAM:

- O arquivo de saída do **bwa mem** é no formato SAM. Converta-o para o formato BAM, mais compacto e eficiente.
- `samtools sort alinhamento.sam > alinhamento_sorted.bam`

### 4. Ordenar e Indexar o Arquivo BAM:

- Ordene o arquivo BAM para facilitar a busca e indexe-o para melhorar o desempenho de ferramentas subsequentes.
- `samtools index alinhamento_sorted.bam`

### 5. Remoção de Duplicatas (opcional):

- Dependendo da aplicação, você pode querer remover duplicatas do seu arquivo BAM para evitar viés em análises subsequentes.
- `samtools rmdup alinhamento_sorted.bam alinhamento_nodups.bam`

### 6. Visualização do Alinhamento:

- Use o IGV (Integrative Genomics Viewer) para visualizar o alinhamento e verificar sua qualidade.

### 7. Avaliação de Cobertura:

- Utilize o comando **samtools depth** para calcular a cobertura do genoma e avaliar a profundidade de sequenciamento em diferentes regiões.
- `samtools depth alinhamento_sorted.bam > cobertura.txt`

# Dia 3 - Genotipagem

## Arquivos

## Atividades

### 1. Preparação do Ambiente:

- Certifique-se de que o freebayes está instalado no seu ambiente. Você pode instalar o FreeBayes usando gerenciadores de pacotes, como o **conda** ou o **pip**, ou compilando-o a partir do código-fonte.
- `conda install -c bioconda freebayes`

### 2. Indexação do Genoma de Referência (se ainda não estiver indexado):

- Assim como na etapa de alinhamento, o genoma de referência deve ser indexado.
- `samtools faidx reference_genome.fa`

### 3. Chamada de Variantes com freebayes:

- Execute o freebayes para chamar variantes a partir do arquivo BAM gerado após o alinhamento.
- `freebayes -f reference_genome.fa -L bam_list.txt -t bed_file.bed > variantes.vcf`

#### Otimizando o uso do freebayes

Caso o computador que esteja utilizando abortar o processo por falta de memória, você pode optar por reduzir o número de amostras em `bam_list.txt`. É importante destacar que a análise correta de genotipagem via **freebayes** requer que todas as amostras sejam analisadas simultaneamente, mas para fim de aprendizado esta é uma estratégia.

### 4. Filtragem de Variantes (opcional):

- Dependendo dos seus critérios e do tipo de análise, pode ser necessário filtrar as variantes chamadas pelo freebayes para reduzir o número de falsos positivos.
- `bcftools view --exclude 'QUAL>5' variantes.vcf > variantes_filtradas.vcf`

Adapte os critérios de filtragem conforme necessário.

#### **5. Análise e Interpretação de Variantes:**

- Utilize ferramentas como `VCFtools` ou `bcftools` para realizar análises adicionais no arquivo VCF, como filtragem específica, anotações e interpretação biológica.

#### **6. Visualização de Variantes:**

- Use o IGV para visualizar as variantes em relação ao genoma de referência e avaliar sua qualidade.

## Dia 4 - Análise de sequenciamento Oxford Nanopore

### ! Importante

O software `guppy` só está disponível para download via comunidade da Oxford Nanopore. Para este tutorial fornecemos um arquivo `.tar` para instalação em sua máquina. Para instalar siga os seguintes passos:

Acesse a pasta que contem o arquivo `.tar` e descompacte;

```
tar -xf ont-guppy-cpu_6.5.7_linux64.tar.gz
```

Verifique o caminho completo para a pasta

```
pwd
```

Executando `guppy` via caminho completo (Exemplo pedindo ajuda)

```
./guppy_basecaller --help
```

Este tutorial foi inspirado...

### Atividades

- Verifique os workflows disponíveis para esta versão de `guppy`;
- Realize uma chamada de base para todos os arquivos `fast5` contidos na pasta ...;

### ⚠ Aviso

O tempo de execução da chama de base (para este conjunto de dados) é superior a **12 horas** em máquinas de uso pessoal, e pode acabar inutilizando seu uso. Caso deseje praticar, recomendamos que seja feito em um momento onde não precise da máquina para outras atividades. Para este tutorial utilize o resultado da chamada de base contido na pasta ...

- Avalie a qualidade do arquivo resultado `.fastq` utilizando `FASTQC` e `NanoPlot`;
- Filtre a read baseado no tamanho e na qualidade utilizando `chopper`;
- Monte o genoma utilizando `BWA` e `minimap2`;



### Opcional

Realize uma montagem *de novo* utilizando o pipeline minimap2-miniasm. Este processo pode levar várias horas/dias. Tome cuidado!

## **Dia 5 - Genotipagem de STRs a partir de dados de NGS**

# Referências

- Bolger, Anthony M., Marc Lohse, and Bjoern Usadel. 2014. “Trimmomatic: A Flexible Trimmer for Illumina Sequence Data.” *Bioinformatics* 30 (15): 2114–20. <https://doi.org/10.1093/bioinformatics/btu170>.
- Danecek, Petr, Adam Auton, Goncalo Abecasis, Cornelis A. Albers, Eric Banks, Mark A. DePristo, Robert E. Handsaker, et al. 2011. “The Variant Call Format and VCFtools.” *Bioinformatics* 27 (15): 2156–58. <https://doi.org/10.1093/bioinformatics/btr330>.
- Danecek, Petr, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, et al. 2021. “Twelve Years of SAMtools and BCFtools.” *GigaScience* 10 (2). <https://doi.org/10.1093/gigascience/giab008>.
- De Coster, Wouter, and Rosa Rademakers. 2023. “NanoPack2: Population-Scale Evaluation of Long-Read Sequencing Data.” Edited by Can Alkan. *Bioinformatics* 39 (5). <https://doi.org/10.1093/bioinformatics/btad311>.
- Garrison, Erik, and Gabor Marth. 2012. “Haplotype-Based Variant Detection from Short-Read Sequencing.” <https://doi.org/10.48550/ARXIV.1207.3907>.
- Li, Heng. 2013. “Aligning Sequence Reads, Clone Sequences and Assembly Contigs with BWA-MEM.” <https://doi.org/10.48550/ARXIV.1303.3997>.
- . 2018. “Minimap2: Pairwise Alignment for Nucleotide Sequences.” Edited by Inanc Birol. *Bioinformatics* 34 (18): 3094–3100. <https://doi.org/10.1093/bioinformatics/bty191>.
- . 2021. “New Strategies to Improve Minimap2 Alignment Accuracy.” Edited by Can Alkan. *Bioinformatics* 37 (23): 4572–74. <https://doi.org/10.1093/bioinformatics/btab705>.
- Martin, Marcel, Murray Patterson, Shilpa Garg, Sarah O Fischer, Nadia Pisanti, Gunnar W Klau, Alexander Schöenhuth, and Tobias Marschall. 2016. “WhatsHap: Fast and Accurate Read-Based Phasing.” <http://dx.doi.org/10.1101/085050>.
- Robinson, James T, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. 2011. “Integrative Genomics Viewer.” *Nature Biotechnology* 29 (1): 24–26. <https://doi.org/10.1038/nbt.1754>.

# ANEXO: Comandos Básicos do Ubuntu via WSL

## Navegação e Diretórios

### Listar Conteúdo do Diretório

```
ls
```

### Mudar de Diretório

```
cd nome_do_diretorio
```

### Diretório Atual

```
pwd
```

### Criar Diretório

```
mkdir nome_do_novo_diretorio
```

## Manipulação de Arquivos

### Copiar Arquivo

```
cp arquivo_origem destino
```

### Mover/Renomear Arquivo

```
mv arquivo_origem novo_nome_ou_destino
```

## **Remover Arquivo**

```
rm nome_do_arquivo
```

## **Visualização de Conteúdo**

### **Visualizar Conteúdo do Arquivo**

```
cat nome_do_arquivo
```

### **Visualizar Conteúdo do Arquivo (página por página)**

```
less nome_do_arquivo
```

## **Pacotes e Atualizações**

### **Atualizar Lista de Pacotes**

```
sudo apt update
```

### **Atualizar Pacotes Instalados**

```
sudo apt upgrade
```

### **Instalar Novo Pacote**

```
sudo apt install nome_do_pacote
```

## **Gerenciamento de Usuários**

### **Adicionar Usuário ao Grupo**

```
sudo usermod -aG nome_do_grupo nome_do_usuario
```

## **Mudar Senha do Usuário**

```
passwd nome_do_usuario
```

## **Processos**

### **Listar Processos**

```
ps aux
```

### **Matar um Processo por ID**

```
kill -9 processo_id
```

### **Monitorar Recursos do Sistema (htop)**

```
htop
```

### **Medir Tempo de Execução de um Comando (time)**

```
time comando_a_ser_medido
```

## **Rede**

### **Verificar Configurações de Rede**

```
ifconfig
```

### **Testar Conexão com um Endereço IP**

```
ping endereco_ip
```

## Entrada e Saída Padrão (stdin/stdout)

- **stdin (Standard Input):** É a entrada padrão de dados. Um programa pode ler dados a partir do stdin. Exemplo:
  - `cat < nome_do_arquivo`
- **stdout (Standard Output):** É a saída padrão de dados. Um programa geralmente imprime resultados no stdout. Exemplo:
  - `ls > lista_de_arquivos.txt`
- **stderr (Standard Error):** É a saída padrão para mensagens de erro. Exemplo:
  - `comando_inexistente 2> erro.log`

## Outros Comandos Úteis

### Ajuda sobre um Comando

`man nome_do_comando`

### Sair do Terminal

`exit`