Aula prática: Montagem de genoma (long + short reads)

1. Instalando programas
2. Organização dos dados
3. Qualidade dos reads (QC)
4. Profiling do genoma
5. Montagem com long reads
6. Avaliação da montagem

1. Instalando programas

Todos os programas serão instalados em um **ambiente conda**. Na pasta “Data” que vocês receberam, tem um arquivo chamado “genome_assembly_course.yml” que contém as instruções para a criação desse ambiente. Simplesmente abra o terminal Linux (lembre-se de navegar até a pasta Data) e cole:

```
conda env create -f genome_assembly_course.yml
```

Agora basta ativar o ambiente:

```
conda activate genome_assembly_course
```

2. Organização dos Dados

É importantíssimo tomar cuidado com a organização dos dados! Se vocês desziparam a pasta “Data” enviada na disciplina, verão que ela contém algumas subpastas, incluindo a subpasta “subsampled”, que contém os arquivos usados durante aula. Vamos agora usar o terminal para fazer pastas onde iremos guardar os resultados das análises.

```
mkdir -p results/{qc,profiling,assembly,polish,eval}
```

3. Qualidade dos reads

O primeiro passo em qualquer workflow de montagem é sempre a checagem da qualidade do sequenciamento. Estamos usando reads de *Arabidopsis thaliana* obtidos por duas diferentes formas: reads longos Nanopore e reads curtos Illumina. Usaremos softwares distintos para cada.

3.1 Long reads – NanoPlot

```
NanoPlot \
  --fastq subsampled/Long/at_hifi_30x.fastq \
  -o results/qc/nanoplot \
```

```
  --threads 4
```

Arquivos para checar (results/qc/nanoplot/):

- LengthvsQualityScatterPlot_dot.png
- NanoStats.txt

3.2 Short reads – FastQC

```
fastqc subsampled/Short/ind_1/SRR1560657_1_sub.fastq.gz
subsampled/Short/ind_1/SRR1560657_2_sub.fastq.gz \
      subsampled/Short/ind_2/SRR1581142_1_sub.fastq.gz
subsampled/Short/ind_2/SRR1581142_2_sub.fastq.gz \
      -o results/qc/fastqc -t 4

multiqc results/qc -o results/qc/multiqc
```

Arquivos para checar (results/qc/fastqc e results/qc/multiqc):

- Relatórios em formato .html

4. Profiling do genoma

4.1 Contagem de k-mers (Jellyfish)

O “profiling” ou perfilamento é uma etapa importante que permite investigar atributos importantes do genoma por meio de reads curtos, utilizando o princípio da contagem de k-mers. Aqui nós usaremos o **jellyfish** para dividir os dados do indivíduo 1 (short reads) em k-mers de 21 pares de base e o **genomescope** para estimarmos heterozigosidade e repetitividade do genoma.

```
#Contando os k-mers com o Jellyfish
jellyfish count \
  -C -m 21 -s 1G -t 4 \
  <(zcat subsampled/Short/ind_1/SRR1560657_1_sub.fastq.gz
subsampled/Short/ind_1/SRR1560657_2_sub.fastq.gz) \
  -o results/profiling/reads.jf

#Criando os dados brutos par aum histograma com o Jellyfish
jellyfish histo -t 4 results/profiling/reads.jf \
  > results/profiling/kmer.histo

#Plotando um histograma e estimando parâmetros com o genomescope2

genomescope2 \

  -i results/profiling/long/kmer.histo \

  -o results/profiling/long/genomescope \

  -k 21
```

Arquivos para checar (results/profiling/genomescope/):

- summary.txt
- transformed_linear_plot.png

5. Montagem com long reads

Como estamos lidando com reads longos sequenciados na plataforma PacBio HiFi, o padrão ouro é usar o **hifiasm** como montador. Este passo é demorado e intensivo computacionalmente. Podem deixar rodando e podemos investigar os resultados na pasta SPOILERS.

5.1 Montagem com hifiasm

```
hifiasm \

  -o results/assembly/at \

  -t 8 \

  subsampled/Long/at_hifi_30x.fastq
```

O hifiasm monta duas haplofases, logo seguiremos analisando ambas e no fim podemos definir com qual seguiremos. O arquivo de saída é o GFA, mas podemos converter para fasta:

```
awk '/^S/{print ">"$2"\n"$3}' results/assembly/at.bp.hap1.p_ctg.gfa > results/assembly/hap1.fasta

awk '/^S/{print ">"$2"\n"$3}' results/assembly/at.bp.hap2.p_ctg.gfa > results/assembly/hap2.fasta
```

Arquivos para checar no IGV:

- hap1.fasta e hap2.fasta
- short_summary.specific.embryophyta_odb10.busco_hap1.txt
- short_summary.specific.embryophyta_odb10.busco_hap2.txt

6. Avaliação da montagem

6.1 Estatísticas básicas – Seqkit

O comando stats do Seqkit nos dá uma visão geral do sequenciamento. Para os próximos exemplos, vamos também comparar com um genoma montado de *A. thaliana* disponível na literatura, que está na pasta Data/reference.

```
seqkit stats results/assembly/hap1.fasta results/assembly/hap2.fasta

seqkit stats reference/A_thaliana_ref.fasta
```

6.2 Estatísticas básicas - QUAST

O QUAST retorna várias estatísticas e plots diagnósticos da montagem.

```
quast results/assembly/hap1.fasta \
    -o results/eval/quast_1

quast results/assembly/hap2.fasta \
```

```
    -o results/eval/quast_2

quast reference/A_thaliana_ref.fasta \
    -o results/eval/quast_ref
```

Arquivos para checar (results/assembly/busco):

- report.pdf em results/eval/quast_1, quast_2 e quast_ref

6.3 Estatísticas básicas - BUSCO

O BUSCO vai avaliar a completude do genoma, buscando encontrar conjuntos de genes ortólogos comuns a grupos específicos (no nosso caso, Embryophyta).

```
busco -i results/assembly/hap1.fasta -l embryophyta_odb10 -m genome -o
results/assembly/busco/busco_hap1 -c 8

busco -i results/assembly/hap2.fasta -l embryophyta_odb10 -m genome -o
results/assembly/busco/busco_hap2 -c 8

busco -i reference/A_thaliana_ref.fasta -l embryophyta_odb10 -m genome -o
results/assembly/busco/busco_ref -c 8
```

Agora vamos isolar os 5 maiores contigs (teoricamente representando os 5 maiores cromossomos) de todos os genomas e rodar novos seqkit stats e busco.

```
seqkit sort -l -r results/assembly/hap1.fasta | \

seqkit head -n 5 \

> results/assembly/hap1_top5.fasta


seqkit sort -l -r results/assembly/hap2.fasta | \

seqkit head -n 5 \

> results/assembly/hap2_top5.fasta


seqkit sort -l -r reference/A_thaliana_ref.fasta | \

seqkit head -n 5 \

> results/assembly/ref_top5.fasta


seqkit stats results/assembly/hap1_top5.fasta results/assembly/hap2_top5.fasta

seqkit stats results/assembly/ref_top5.fasta


busco -i results/assembly/hap1_top5.fasta -l embryophyta_odb10 -m genome -o
results/assembly/busco/busco_hap1_top5 -c 8
```

busco -i results/assembly/hap2_top5.fasta -l embryophyta_odb10 -m genome -o results/assembly/busco/busco_hap2_top5 -c 8

busco -i reference/A_thaliana_ref_top5.fasta -l embryophyta_odb10 -m genome -o results/assembly/busco/busco_ref_top5 -c 8

Arquivos para checar (results/assembly/busco):

- short_summary.specific.embryophyta_odb10.busco_ref.txt em todas as pastas