

Jun 10, 2024

Version 1

## Montagem de genomas de fungos (short reads) V.1

DOI

<https://dx.doi.org/10.17504/protocols.io.261ge5rpog47/v1>

Thiago Mafra Batista<sup>1</sup>

<sup>1</sup>Universidade Federal do Sul da Bahia



Thiago Mafra Batista

Universidade Federal do Sul da Bahia

### Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



**DOI:** <https://dx.doi.org/10.17504/protocols.io.261ge5rpog47/v1>

**Protocol Citation:** Thiago Mafra Batista 2024. Montagem de genomas de fungos (short reads). protocols.io  
<https://dx.doi.org/10.17504/protocols.io.261ge5rpog47/v1>

**License:** This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

**Protocol status:** In development

We are still developing and optimizing this protocol

**Created:** June 07, 2024



**Last Modified:** June 10, 2024

**Protocol Integer ID:** 101378

**Keywords:** genome assembly, short reads, fungal nuclear genome, bacterial genome database, nuclear genome, genome, illumina short reads with spades software, short read, illumina short read, fungal, de genoma, busco software, read, fungo

## Abstract

This protocol provides step-by-step instructions for students and researchers to assemble fungal nuclear genomes using Illumina short reads with SPAdes software. Before assembling the genome, we will align the short reads against a bacterial genome database to eliminate possible contaminations. Finally, we will evaluate the completeness of the assemblies using the BUSCO software.

## Materials

Softwares necessários para toda pipeline:

1. Fastqc (<https://github.com/s-andrews/FastQC>)
2. Multiqc (<https://github.com/MultiQC/MultiQC>)
3. BWA (<https://github.com/lh3/bwa>)
4. SPAdes v.3.15.5 (<https://github.com/ablab/spades>)
5. BUSCO v5.4.7 (<https://busco.ezlab.org/>)

Scripts ou softwares acessórios:

- scaffold\_stats.pl ([https://github.com/sujaikumar/assemblage/blob/master/scaffold\\_stats.pl](https://github.com/sujaikumar/assemblage/blob/master/scaffold_stats.pl))
- bioawk (<https://github.com/lh3/bioawk>)



## Avaliação da qualidade do sequenciamento

- 1 Para avaliarmos os parâmetros de qualidade, sobretudo o phred score, utilizaremos dois softwares: **FastQC** e **MultiQC**.

```
fastqc -t 64 *.fastq -o .
```

O fastqc gera um arquivo .html e um arquivo .zip para cada read. Os arquivos .html podem ser abertos em qualquer navegador de internet. Já o multiqc utiliza as informações geradas pelo fastqc para gerar um relatório mais completo. Ele deve ser executado no mesmo diretório onde foram salvos os arquivos de saída do fastqc.

```
multiqc .
```

Será gerado um arquivo chamado multiqc\_report.html que também deve ser visualizado em qualquer navegador de internet.

## Filtragem de possíveis contaminantes bacterianos

- 2 Com o intuito de eliminar possíveis contaminantes bacterianos, vamos filtrar as reads alinhando-as contra um banco de sequências de genomas bacterianos depositadas no RefSeq.

Configure os caminhos das reads e do diretório de trabalho de acordo com o projeto. A seguir, um script em bash que filtra as raw reads contra o banco de dados RefSeq bacteriano. Ao fim, é gerado a contagem de reads pareadas e singletons alinhadas e não alinhadas no RefSeq bacteriano.

```
#!/bin/bash

# Configuração dos caminhos de entrada (reads) e do diretório de
trabalho.
reads="/data/reads/leveduras_inct/abril_24/S727-BRT367"
work_dir="/home/thiagomafra/projetos/leveduras/inct/abril24/brt367
/filtragem"

echo "Alinhando reads vs Refseq Bacteria\n"
# Execução do alinhamento usando BWA MEM. (Linha comentada para
segurança)
bwa mem -t 24 -a -P
/data/databases/refseq/representative_bacteria/all_representatives
_refseq_217_bacteria.fna \
    $reads/S727-BRT367_S382_L001_R1_001.fastq \
    $reads/S727-BRT367_S382_L001_R2_001.fastq >
"$work_dir/BRT367_output_bwa_vs_refseq_bact_217.sam" 2>
"$work_dir/bwa.log"

echo "Alinhamento concluído.\n"

echo "Convertendo SAM para BAM e ordenando..."
# Conversão de SAM para BAM e ordenação dos arquivos. (Linha
comentada para segurança)
samtools view -b
"$work_dir/BRT367_output_bwa_vs_refseq_bact_217.sam" | samtools
sort -@24 -o "$work_dir/BRT367_sorted.bam"

echo "Conversão e ordenação concluídas.\n"

echo "Gerando as reads FastQ para alinhadas, incluindo
singletons."
# Geração de arquivos FastQ para reads alinhadas e singletons.
samtools fastq -@24 -F 4 -1 "$work_dir/BRT367_aligned_R1.fastq" -2
"$work_dir/BRT367_aligned_R2.fastq" \
    -s "$work_dir/BRT367_aligned_singletons.fastq"
"$work_dir/BRT367_sorted.bam"

echo "Gerando as reads FastQ para não alinhadas, incluindo
singletons."
# Geração de arquivos FastQ para reads não alinhadas e singletons.
samtools view -b -f 4 "$work_dir/BRT367_sorted.bam" | \
samtools fastq -@24 -1 "$work_dir/BRT367_unaligned_R1.fastq" -2
"$work_dir/BRT367_unaligned_R2.fastq" \
```

```
-s "$work_dir/BRT367_unaligned_singletons.fastq"

echo "Contando reads..."
# Contagem das reads em cada categoria.
aligned_R1_count=$(grep -c '^@'
"$work_dir/BRT367_aligned_R1.fastq")
aligned_R2_count=$(grep -c '^@'
"$work_dir/BRT367_aligned_R2.fastq")
aligned_singleton_count=$(grep -c '^@'
"$work_dir/BRT367_aligned_singletons.fastq")
unaligned_R1_count=$(grep -c '^@'
"$work_dir/BRT367_unaligned_R1.fastq")
unaligned_R2_count=$(grep -c '^@'
"$work_dir/BRT367_unaligned_R2.fastq")
unaligned_singleton_count=$(grep -c '^@'
"$work_dir/BRT367_unaligned_singletons.fastq")
original_R1_count=$(grep -c '^@' "$reads/S727-
BRT367_S382_L001_R1_001.fastq")
original_R2_count=$(grep -c '^@' "$reads/S727-
BRT367_S382_L001_R2_001.fastq")

echo "Calculando totais e percentual de alinhamento."
# Cálculo dos totais e percentuais de alinhamento, incluindo
singletons.
total_aligned=$((aligned_R1_count + aligned_R2_count +
aligned_singleton_count))
total_unaligned=$((unaligned_R1_count + unaligned_R2_count +
unaligned_singleton_count))
total_processed=$((total_aligned + total_unaligned))
total_original=$((original_R1_count + original_R2_count))
percent_aligned=$(echo "scale=2; $total_aligned * 100 /
$total_original" | bc)

echo "Salvando resultados no arquivo read_counts.txt"
# Salvando os resultados no arquivo de texto.
echo "Reads alinhadas R1: $aligned_R1_count" >
"$work_dir/read_counts.txt"
echo "Reads alinhadas R2: $aligned_R2_count" >>
"$work_dir/read_counts.txt"
echo "Reads singletons alinhadas: $aligned_singleton_count" >>
"$work_dir/read_counts.txt"
echo "Reads não alinhadas R1: $unaligned_R1_count" >>
"$work_dir/read_counts.txt"
echo "Reads não alinhadas R2: $unaligned_R2_count" >>
"$work_dir/read_counts.txt"
```



```
echo "Reads singletons não alinhadas: $unaligned_singleton_count"
>> "$work_dir/read_counts.txt"
echo "Total de reads originais: $total_original" >>
"$work_dir/read_counts.txt"
echo "Total de reads processadas (alinhadas, não alinhadas,
singletons): $total_processed" >> "$work_dir/read_counts.txt"
echo "Percentual de reads alinhadas: ${percent_aligned}%" >>
"$work_dir/read_counts.txt"

cat read_counts.txt
rm *.sam
echo "FIM"
```

## Montagem do genoma com SPAdes

### 3 Instruções:

- Realizar, pelo menos, três diferentes montagens com os dados testando parâmetros como o `--cov-cutoff`; por padrão, ele está off; testar `--cov-cutoff auto` e um valor abaixo de 100, por exemplo `--cov-cutoff 80`
- Sempre habilitar o parâmetro `--isolate` quando houver uma alta cobertura de dados (acima de 100x)
- Utilizar as reads não alinhadas no RefSeq bacteriano
- Tempo estimado para cada montagem: 90 minutos

### 4 Para montar o genoma com o SPAdes, lembre-se de utilizar as reads que não alinharam no RefSeq bacteriano.

```
spades.py -1 ../filtragem/BRT367_unaligned_R1.fastq -2
../filtragem/BRT367_unaligned_R2.fastq --isolate -t 64 -o
spades_run1 --cov-cutoff 80
```

Avaliação das montagens com `scaffold_stats.pl` e BUSCO

O spades gera dois principais arquivos de output, o `contigs.fasta` e o `scaffolds.fasta`. Avalie os parâmetros de avaliação da qualidade da montagem em ambos arquivos e decida qual utilizar a partir das melhores métricas.

O Genbank só aceita contigs que tenham pelo menos 200 pb. Para eliminar os contigs menores que 200 pb utilize o comando abaixo:



```
bioawk -c fastx 'length($seq) >=200{print ">$name"\n"($seq)}'  
contigs.fasta > contigs_200bp.fasta
```

- rodar o script scaffold\_stats.pl com todos os arquivos contigs\_200bp.fasta, assim:

```
scaffold_stats.pl -f contigs_200bp.fasta scaffolds_200bp.fasta -N  
1 -t 20 1000 | tee stats.txt
```

Avalie os seguintes parâmetros:

- Tamanho do genoma montado
- Quantidade de contigs/scaffolds
- N50
- L50
- Maior contig/scaffold

Rodar o BUSCO para avaliar a presença de ortólogos conservados no grupo taxonômico de interesse. Neste caso, saccharomycetes

```
busco -l saccharomycetes --download_path /data/busco_downloads/ -c  
64 --mode geno --in contigs_200bp.fasta -o busco_geno
```

Observação: especifique um local para download do orthodb de saccharomycetes para que nas próximas vezes que rodar o busco não seja necessário um novo download. No caso acima, especifiquei /data/busco\_downloads/. Isso economiza um bom tempo ao rodá-lo.