Nucleotide Sequence Analysis

Class Workbook

Miha Prajs

July 1, 2025

## Environment Preparation

Before we can start analyzing sequences we must set up the environment. In new terminal window using “weget” command we will download Conda Package Manager installation script from official Anaconda repository. Then we run downloaded script using “bash” command followed by script name. Then we will set channels where we will download needed packages from using “conda config --append channels” command. Now it’s time to set working environment with “conda create -n” followed by a environment’s name and activating it with “conda activate” and it’s name. Then we can install all the necessary packages using “conda install”. It is also recommended that you regularly update installed packages[[1]](#footnote-20).

# Installation of Miniconda   
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh  
bash ./Miniconda3-latest-Linux-x86\_64.sh  
  
conda config --append channels defaults  
conda config --append channels bioconda  
conda config --append channels conda-forge  
  
# Environment preparation  
conda create -n anz  
conda activate anz  
   
# Installation of necessary tools  
conda install emboss entrez-direct sra-tools  
  
# Update instruction  
conda update --all # follow the instructions in terminal window

After the initial set up is complete we can start downloading and analyzing sequences.

## Exercise 1

Download and check quality of the sequence. We first installed additional tools, then we downloaded the sequences and created a quality reporsts.

conda install fastqc multiqc  
  
prefetch SRR30833064  
fasterq-dump SRR30833064  
  
prefetch SRR30833065  
fasterq-dump SRR30833065  
  
fastqc SRR30833064.fastq  
fastqc SRR30833065.fastq  
multiqc .

### Questions:

1. What is the aim of the study, described in the scientific article? *The aim of the study is to investigate how W. pipientis affects germline stem cell function and fertility in fruit fly.*
2. Provide some info about the dataset you will work with and which sequencing technology was used. *RNA-seq datasets are generated from the ovaries of fruit fly.*
3. Which kit was used for sequencing library preparation? Does this kit preserve strand information (stranded library) or not? *Sequencing was done using Ilumina NextSeq500. It can preserve strand information.*
4. What is the advantage of stranded mRNA library preparation compared to non-stranded library? *They retain information about transcriptional directions of RNA, have better resolution, are more accurate.*
5. How is a fastq file composed? *FASTQ file is a text-based format. It contains multiple reads from high-throughput sequencing machines. Each read is composed of 4 lines (line 1 is seq. identifier, line 2 is raw sequence, line 3 is only a +, line 4 contaions quality score).*
6. How can I count the number of reads in a fastq file? Describe different ways to perform that. *Reads can be counted using multiple linux commands, like grep, awk, wc.*
7. What about the quality of your reads? *We can use tools like FastQC and MultiQC (if we have multiple fastq files).*
8. Describe your fastqc and/or multiqc and interpret the results. *FastQC reports show sequence quality per base, duplication levels, and adapter content of a single sequence while MultiQC combines results of multiple sequences into a single report.*

## Exercise 2

Using fastp tool we checked the presence of primer, adapter and remove bases with low quality score A.K.A. trimmed the sequence. After checking for quality filtering we ran fastqc again to check if the quality of sequence improved. After multiple runs and changes the best result was used for further analysis.

conda install fastp  
  
fastp -i SRR30833064.fastq -o SRR30833064\_trimmed.fastq \  
 -t 5 -x --poly\_x\_min\_len=7 --dedup --length\_limit=80  
   
fastqc SRR30833064\_trimmed.fastq

After trimming the quality was better in some metrics and a little worse in others. Then we downloaded the reference genome form NCBI, build index from it and aligned the reads using hisat2 program.

1. Download genome and build index:

datasets download genome accession GCF\_007971685.1 --include genome,gff3,gtf,cds  
  
unzip ncbi\_dataset.zip  
  
hisat2-build ncbi\_dataset/data/GCF\_000001215.4/GCF\_000001215.4\_Release\_6\_plus\_ISO1\_MT\_genomic.fna genome

1. Align reads:

hisat2 -x genome -U SRR30833064\_trimmed.fastq -p 4 -S SRR30833064.sam

After aligning we quantified expression of transcript using Salmon:

salmon index -t ncbi\_dataset/data/GCF\_000001215.4/cds\_from\_genomic.fna -i index\_dir  
  
  
salmon quant -i index\_dir/ -l A -1 SRR30833064.fastq -2 \  
 SRR30833064\_trimmed.fastq -o output\_dir

Uniquely mapped reads were 323939 or 82.39 % and 21958 or 5.58 % were mapped multiple times.

### Questions:

1. Describe the GTF. *It’s a tab-delimited file describing gene annotations on given genome. It’s a refined version of a GFF file.*

2.Examine GTF files. Which information can be found in these files? *GTF file represents a genomic feature. It includes next fields: seqname, source, feature, start, end, score, strand frame and attribute.*

3.How many genes are present? *Using command*

grep -w "gene" ncbi\_dataset/data/GCF\_000001215.4/genomic.gtf | awk -F 'gene\_id "' '{print $2}' | cut -d '"' -f1 | sort | uniq | wc -l

*it found 17872 genes.*

1. Provide me the commands and results for counting the number of sequences in the various fasta files (DNA, RNA, protein). *Command:*

for file in ncbi\_dataset/data/GCF\_000001215.4/\*.fna; do echo -n "$file: "; grep -c "^>" "$file"; done  
for file in ncbi\_dataset/data/GCF\_000001215.4/\*.faa; do echo -n "$file: "; grep -c "^>" "$file"; done

*Result:*

ncbi\_dataset/data/GCF\_000001215.4/cds\_from\_genomic.fna: 30802  
ncbi\_dataset/data/GCF\_000001215.4/GCF\_000001215.4\_Release\_6\_plus\_ISO1\_MT\_genomic.fna: 1870  
ncbi\_dataset/data/GCF\_000001215.4/rna.fna: 34526  
ncbi\_dataset/data/GCF\_000001215.4/protein.faa: 30802

1. Describe differences between different genomic fasta files. *Genome fasta file contains whole genome (chromosomes or scaffolds), CDS only contains coding seq., protein fasta contain protein seq., RNA/transcriptome fasta includes mRNA or cDNA seq.*

## Exercise 3

After producing the alignment, reads are in random order with respect to their position in the reference genome. To examine the alignment we sorted and indexed them.

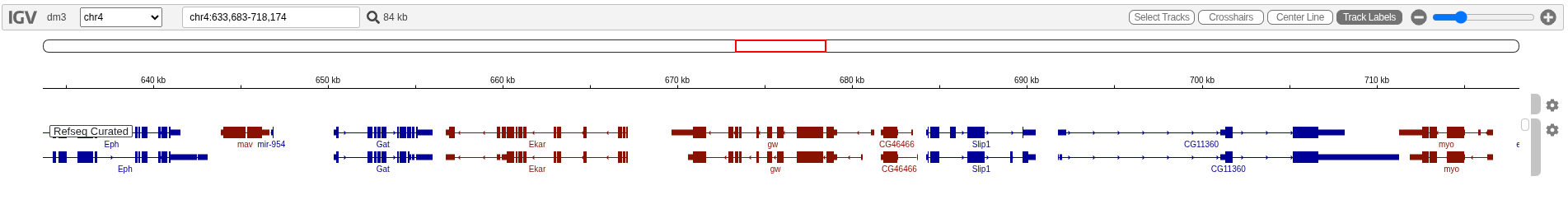
1. Convert and sort SAM files:

conda install samtools tablet  
  
samtools view -o SRR30833064.bam SRR30833064.sam  
  
samtools sort -o SRR30833064\_sorted.bam SRR30833064.bam  
  
samtools index SRR30833064\_sorted.bam

1. Create index of genome:

samtools faidx ncbi\_dataset/data/GCF\_000001215.4/GCF\_000001215.4\_Release\_6\_plus\_ISO1\_MT\_genomic.fna   
  
samtools flagstat SRR30833064.sam  
  
samtools flags 163

To visualize and check if sequences are sorted there is a tool on web named [IGV](igv.org) that can do that.



This picture shows alignmet of our RNA to a RefSeq’s sequence.

Then we used featureCounts tool to count aligned reads per gene. Before that we checked if dataset is strand specific using infer\_experiment.py from RSeQC package.

conda install bedops rseqc subread  
  
gtf2bed < ncbi\_dataset/data/GCF\_000001215.4/genomic.gtf > genomic.bed  
  
infer\_experiment.py -r genomic.bed -i SRR30833064\_sorted.bam

Results of infer\_experiment.py:

Loading SAM/BAM file … Total 200000 usable reads were sampled

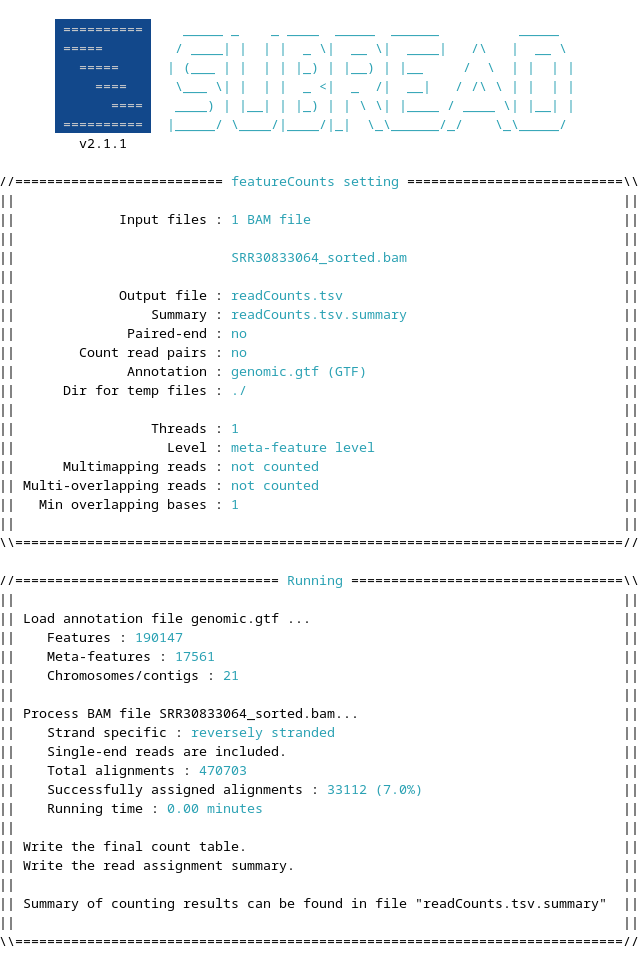
This is SingleEnd Data Fraction of reads failed to determine: 0.1484 Fraction of reads explained by “++,–”: 0.8392 Fraction of reads explained by “+-,-+”: 0.0124

83.92 % of genes were mapped on the same strand as their parental gene, 1.24 % of them were mapped on the opposite strand and 14.84 % were not mapped.

Then we counted fragments per gene:

featureCounts -a ncbi\_dataset/data/GCF\_000001215.4/genomic.gtf -o \  
 readCounts.tsv -s 2 SRR30833064\_sorted.bam

Output:



The 10 most expressed genes are (in order from most to least): Dmel\_CG42600, Dmel\_CG1896, Dmel\_CG32095, Dmel\_CG8616, Dmel\_CG17272, Dmel\_CG3504, Dmel\_CG8657, Dmel\_CG3195, Dmel\_CG14966, Dmel\_CR43812.

## Exercise 4 and 5

Most of the tasks done in this lab report can also be done using internet tools, like Galaxy. Using [galaxy](usegalaxy.eu) we repeat everything we have done using just linux commands while only difference was that we used different dataset which contained double stranded sequence. We trimmed it using tool “Trim Galore!” ran FastQC, assembled with SPAdes, evaluated with QUAST. Then we evaluated again using BUSCO. After that we downloaded scaffolds fasta file and preform evaluation and completeness of microbial genome assemblies using CheckM. With the CheckM we got a completeness percentage and contamination percentage. In my case completeness was 99.08 % and on 0.49 % of the sequence was contaminated.

### Questions

1. What is N50? And L50? Describe these parameters. *N50 is a length-weighted median statistic that tells you how contiguous your assembly is.*
2. What’s a kmer? In which analysis could be used? How many contigs have you obtained? *A kmer si a sequence of nucleotides with lenght of k. It is used in the prooces of assembling the genome. According to file contigs.fna there is a 221 contigs.*
3. Provide BUSCO results.

| |Results from dataset bacteria\_odb10 | |
| --- |
| |C:97.6%[S:97.6%,D:0.0%],F:2.4%,M:0.0%,n:124 | |121 Complete BUSCOs (C) | |121 Complete and single-copy BUSCOs (S) | |0 Complete and duplicated BUSCOs (D) | |3 Fragmented BUSCOs (F) | |0 Missing BUSCOs (M) | |124 Total BUSCO groups searched | |

1. How many genes were retrieved in the gff3 of PROKKA output. How can you count that? Provide some of the results. *PROKKA retrieved 3418 gens. It is part of the output txt.*

## Exercise 6

Last thing done is to document the process. To document the code and to make sure that everyone uses the same tools and commands it is easiest to crate a snakemake file or a bash script with all the code.

# Snakemake  
  
samples = []  
genome = "/path/to/reference.fasta"  
  
with open('list\_of\_samples.txt') as f:  
 samples = f.read().splitlines()  
  
rule all:  
 input:  
 expand("mapped/{sample}sorted.mapping.bam", sample=samples),  
 expand("report/flagstat/{sample}sorted.mapping.flagstat.txt", sample=samples)  
  
rule trimming:  
 input:  
 read1="{sample}.fastq.gz"  
 output:  
 trimmed="trimmed/{sample}.trimmed.fastq.gz",  
 report="report/fastp/{sample}.html"  
 log:  
 log="logs/{sample}.fastp.log"  
 conda:  
 "fastp"  
 group: "pipeline"  
 shell:  
 "fastp -i {input.read1} -o {output.trimmed} -V -w 16 -x -g -n 2 -5 -3 -p -l 75 -M 26 -h {output.report}"  
  
rule mapping:  
 input:  
 reference=genome,  
 trimmed="trimmed/{sample}.trimmed.fastq.gz"  
  
 output:  
 "mapped/{sample}mapping.bam"  
 conda:  
 "alignment"  
 group: "pipeline"  
 shell:  
 "bwa mem -t 4 -M {input.reference} {input.trimmed} | samtools view -@ 4 -bS - > {output}"  
  
rule sorting:  
 input:  
 mapping="mapped/{sample}mapping.bam"  
 output:  
 sorted="mapped/{sample}sorted.mapping.bam"  
 conda:  
 "alignment"  
 group:  
 "pipeline"  
 shell:  
 "samtools sort -@ 24 -o {output.sorted} {input.mapping}"  
  
rule flagstat:  
 input:  
 "mapped/{sample}sorted.mapping.bam"  
 output:  
 "report/flagstat/{sample}sorted.mapping.flagstat.txt"  
 conda:  
 "alignment"  
 group:  
 "pipeline"  
 shell:  
 "samtools flagstat -@ 24 {input} > {output}"  
   
rule unmapped:  
 input:  
 "mapped/{sample}sorted.mapping.bam"  
 output:  
 "mapped/{sample}\_unmapped.bam"  
 conda:  
 "alignment"  
 group:  
 "pipeline"  
 shell:  
 "samtools view -u -f 12 -F 256 {input} > {output}"

1. Please note that in this workbook we will only use bash code [↑](#footnote-ref-20)