CINEMA Bioinformatics Practical

Background

- A clinical collaborator is carrying out a study of bacterial vaginosis (BV).

- BV is the most common vaginal infection in women of reproductive age. BV is often treatment-refractory and characterized by a shift in the vaginal microbiome.

- Our collaborator has collected 6 samples of women with BV and 6 samples of women without BV, followed by full-length amplification of the 16S gene and Oxford Nanopore sequencing.

- Our job is to carry out an exploratory analysis of the data and develop hypotheses about the nature of the BV-associated shift of the vaginal microbiome.

Data package

- Obtain the data package from:

  https://www.dropbox.com/s/hw9zrwoxbddsdf2/CINEMADataPackage.zip?dl=0

- The ZIP archive will contain 12 FASTQ files (from our patients) and a file (SampleMapping.txt) that contains the information which files come from patients with or without vaginosis (controls).

Install Emu

- We will use Emu to analyze the Nanopore 16S data

- As a first step, you need to install Emu and its default database.

- Installation instructions are here: https://gitlab.com/treangenlab/emu
  - If you have issues with the OSF client, use this database download link instead: https://www.dropbox.com/scl/fi/vo4g19eqc1oq8lfb6o58m/emu.tar?rlkey=9m4n9s1miqo6da060xfoz2nxk&dl=0

- If using Windows, you probably want to use the Windows Subsystem for Linux
  - Install WSL: https://learn.microsoft.com/en-us/windows/wsl/install
  - Once you have WSL installed, install Conda: https://docs.conda.io/en/latest/miniconda.html
  - E.g. like that (within a WSL shell):

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
chmod u+x Miniconda3-latest-Linux-x86_64.sh
./Miniconda3-latest-Linux-x86_64.sh
```

- Make sure you have a functional Emu by typing „emu –help" and checking that the Emu help message appears.

Run Emu on the data

- The basic syntax of calling Emu is:

  ```
  emu abundance --type map-ont --db DB file.fastq
  ```

  (… make sure that your database is actually installed in directory `DB`)

- This will by default produce a file `results/file_rel-abundance.tsv`

- You can run Emu one file at a time, but you can also use e.g. Bash to process multiple files in one go – like that:

  ```
  for file in Data/*.fastq; do emu abundance --type map-ont --db DB "${file}"; done
  ```

  … this command assumes that your FASTQ files are in a directory called `Data`.

Task

- Apply Emu to the downloaded data package files

Merge results files

- For 12 input FASTQ files, we get 12 output `.tsv` files.

- You can merge these using the script `merge_Emu_results.py` (from here: https://www.dropbox.com/scl/fi/qnhzy2ptidnszmrymma70/merge_Emu_results.py?rlkey=9znyorfuub1f9rq4cmfzqoheu&dl=0)

- Example syntax:

```
python3 merge_Emu_results.py results/*.tsv
```

- This will produce a file `output.tsv`.

Exploratory data analysis

- Import the combined .tsv file into a spreadsheet application of your choice.

- Annotate each sample ID column with the type of sample (vaginosis or control).

- Create a heatmap-based visualization of the relative abundances.

- Find out which species are most relevant in the combined sample, e.g. by displaying only rows that correspond to species with >= 10% abundance in any of the analyzed samples.

Analysis in R and significance testing

- Excel and Google Sheets are fine for exploratory data analyses; R (or Python) are, however, much more powerful and also enable formal significance testing.

- Install R and the phyloseq and DESeq2 packages with

```
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install(c("phyloseq"))
BiocManager::install(c("DESeq2"))
```

Analysis in R and significance testing

- As a first step, we want to load the combined output table:

```
library(phyloseq)
library(mia)

D <- read.delim("C:/Users/path/to/output.tsv")
sampleTypes <- read.delim("C:/Users/Alexa/Dropbox/CINEMA Paris Teaching/Data/SampleMapping.txt",
header = F)
otumatrix <- as.matrix(D[c(2:13)]) * 1000 + 1
taxmat <- as.matrix(D[c(14:21)])

rownames(otumatrix) <- D[[1]]
colnames(taxmat) <- c("Species", "Genus", "Family", "Order", "Class", "Phylum", "Clade", "Superkingdom")
rownames(taxmat) <- D[[1]]

taxmat <- taxmat[,rev(colnames(taxmat))]

OTU <- otu_table(otumatrix, taxa_are_rows = TRUE)
TAX <- tax_table(taxmat)
sampledata <- sample_data(data.frame(
  SampleType = sampleTypes[[2]],
  row.names=sampleTypes[[1]],
  stringsAsFactors=FALSE
))

physeq <- phyloseq(OTU, TAX, sampledata)
```

Input: un-normalized read counts and add +1 to avoid numerical degeneracy

If your input files do not all have the same number of reads, this needs to be adapted!

Analysis in R and significance testing

- Let’s do some visualization:

```
plot_bar(physeq, fill = "Family")
plot_bar(physeq, x = "SampleType", fill = "Family")
```

Analysis in R and significance testing

- … and some forrmal significance testing

```
library(DESeq2)

physeqFamily <- tax_glom(physeq, "Family")

deseq_dataset = phyloseq_to_deseq2(physeqFamily, ~ SampleType)
deseq_analysis = DESeq(deseq_dataset, test="Wald", fitType="parametric")

res = results(deseq_analysis, cooksCutoff = FALSE, tidy = TRUE)
stopifnot(rownames(tax_table(physeqFamily)) == res[[1]])
res = cbind(res, tax_table(physeqFamily))

res_small <- res[c("row", "log2FoldChange", "pvalue", "padj", "Family")]
res_small[order(res_small$pvalue),]
```

Task

- Have a look at the results and compare them to the visual analyses;
  also look up Bacterial Vaginsos on Wikipedia. How do our results line up with what is know about BV?