Tutorial 12: Centrifuge

Gavin Hearne
Mohammad Saleh Refahi

Background

What is Centrifuge?

- Centrifuge is a metagenomic abundance estimation tool developed by researchers at JHU that labels and quantifies metagenomic reads
- Most metagenomic classifiers either suffer from either speed (such as Naive Bayes Classifier, PhymmBL, and MegaBLAST), or index size (kraken) issues. This tool aims to solve both of these
- This is done through the use of a novel indexing scheme based on two data structures:
  - Burrows-Wheeler transform
  - Ferragina-Manzini (FM) index

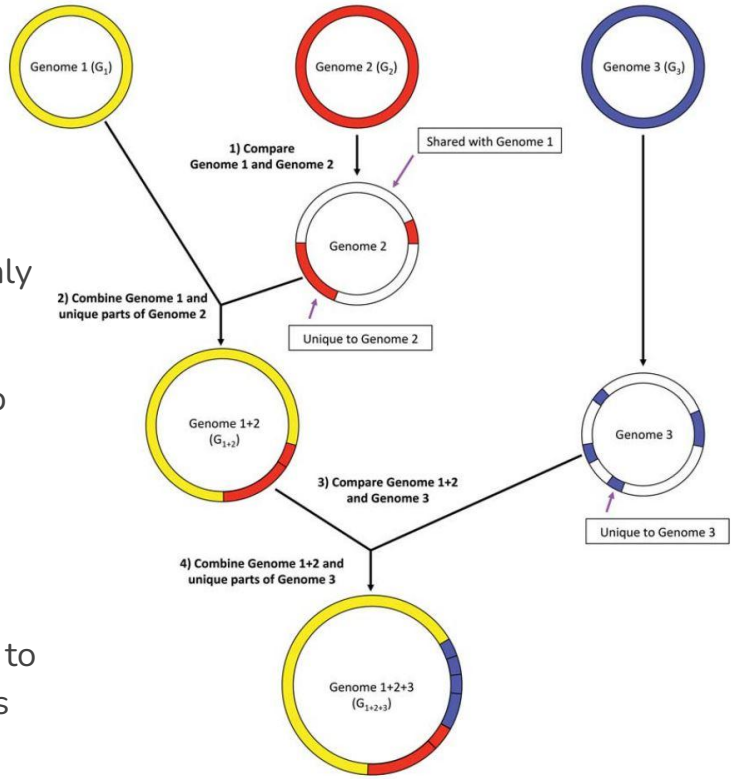
Database sequence compression

Multiple genomes of the same species are compressed by storing near-identical sequences only once.

- This can reduce total sequence lengths by up to 89%.

An FM-index is then created based on these precompressed sequences

- The FM-index is quite small when compared to *k*-mer indexing methods that store all *k*-mers (such as kraken)

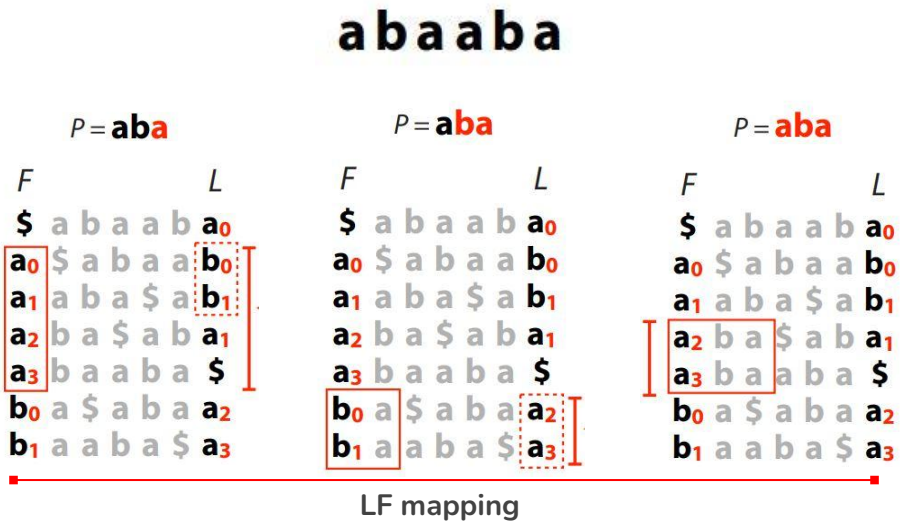


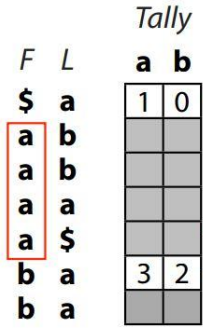
The FM-index

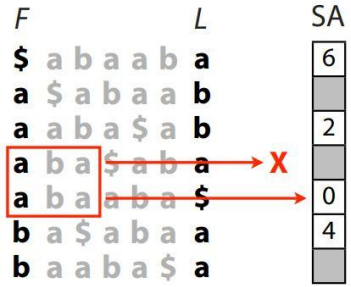
The FM index combines the BTW with a few auxiliary data structures - this consists of 4 components:

- First column (F) of the BW matrix
- Last Column (L) of the BW matrix
- Checkpoints
  - Allows the steps of FM querying to be performed in O(1) time
- Suffix Array
  - Quickly finds where all occurrences of a query appear in the initial sequence

This has the potential to significantly reduce the combined size of the stored genomes, while also allowing for faster queries of the index







Building Indexes

Prebuilt:

The researchers behind Centrifuge have put together three main indexes that are readily available:

- p+h+v: bacterial, human, and viral genomes [~12G]

- p_compressed: bacterial genomes compressed at the species level [~4.2G]

- p_compressed+h+v: combination of the two above [~8G]

Custom:

If these indexes do not suit your needs, it is possible to build custom indexes from arbitrary sequences.

For each sequence, centrifuge needs the following:

- nodes.dmp file from the NCBI taxa dump to build the taxa tree
- sequence ID to taxonomy ID map

Classification

The FM-index allows for Centrifuge to exploit the advantages of both large and small *k*-mers:

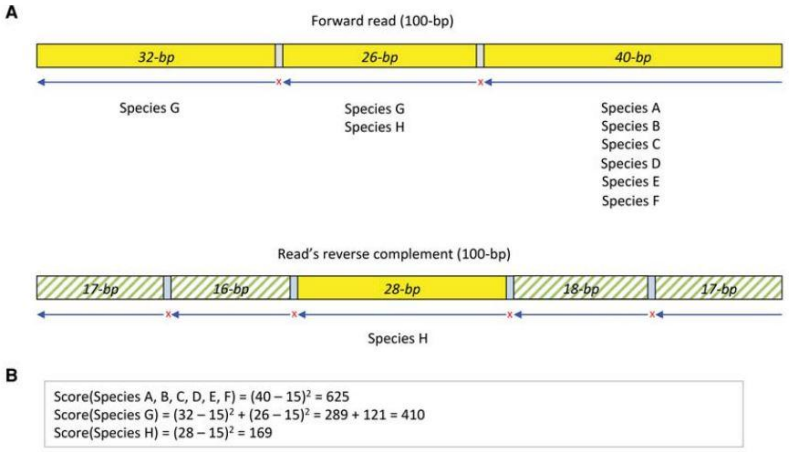
- As *k*-mers size increases, precision is increased but sensitivity is reduced

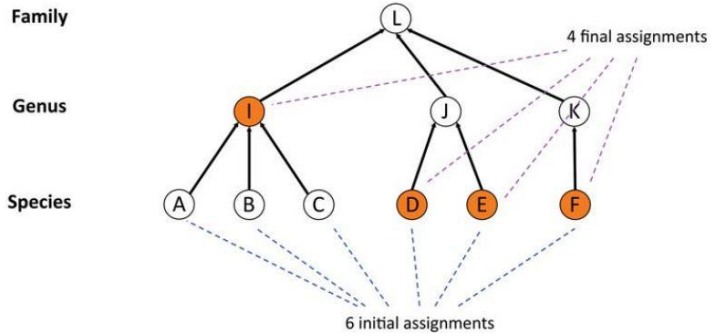
Centrifuge begins by finding 16-bp exact matches, then extending those matches as far as possible. This is performed both with the forward and reverse complement.

Classifications are then performed using only mappings with at least one 22-bp match.

Higher taxonomic level classifications are performed by traversing up the tree and counting number of matches per rank

$$\text{Score}(\text{Species } X) = \sum_{\text{hit} \in \text{Species } X} (\text{length}(\text{hit}) - 15)^2.$$





Abundance analysis

Centrifuge also is capable of performing abundance analysis at any taxonomic rank

- Likelihood for a specific configuration of species abundance α given the read assignments

$$L(\alpha|C) = \prod_{i=1}^{R} \sum_{j=1}^{S} \frac{\alpha_j l_j}{\sum_k^S \alpha_k l_k} C_{ij},$$

To find the abundance that maximizes the likelihood function, the following EM procedure is repeated until the difference between the previous estimate of abundances and the current estimate is less than 10−10

- Expectation (E-step): the estimated number of reads assigned to species j.

$$n_j = \sum_{i=1}^{R} \frac{\alpha_j C_{ij}}{\sum_{k=1}^{S} \alpha_k C_{ik}},$$

- Maximization (M-Step): the updated estimate of species j's abundance

$$\alpha_j' = \frac{n_j/l_j}{\sum_{k=1}^{S} n_k/l_k},$$

Usage

Installing Requirements Packages

one-time set up of Bioconda with the following commands.

```
[ ]: !conda config --add channels defaults
     !conda config --add channels bioconda
     !conda config --add channels conda-forge
     !conda config --set channel_priority strict
```

Install the following packages in conda environments

```
[ ]: # Install mamba in the base conda environment
     # Note: This step might be optional, as mamba is not always necessary
     !conda install -n base --override-channels -c conda-forge mamba 'python_abi=*=*cp*'
```

```
[ ]: # Install the ete3 package using conda
     !conda install ete3 --quiet --yes

     !conda install -c conda-forge biopython

     # Install the centrifuge package from the bioconda channel using conda
     !conda install bioconda::centrifuge


     # Install r-remotes and bioconductor-rsamtools using mamba
     !mamba install --yes --quiet r-remotes bioconductor-rsamtools

     # Install ipywidgets using pip
     !pip install ipywidgets

     !pip install epi2melabs
```

Downloading Centrifuge Index

h+p+v+c: human genome, prokaryotic genomes, and viral genomes including SARS-CoV-2 genomes.

[9]:
```
!wget https://zenodo.org/records/3732127/files/h+p+v+c.tar.gz
```

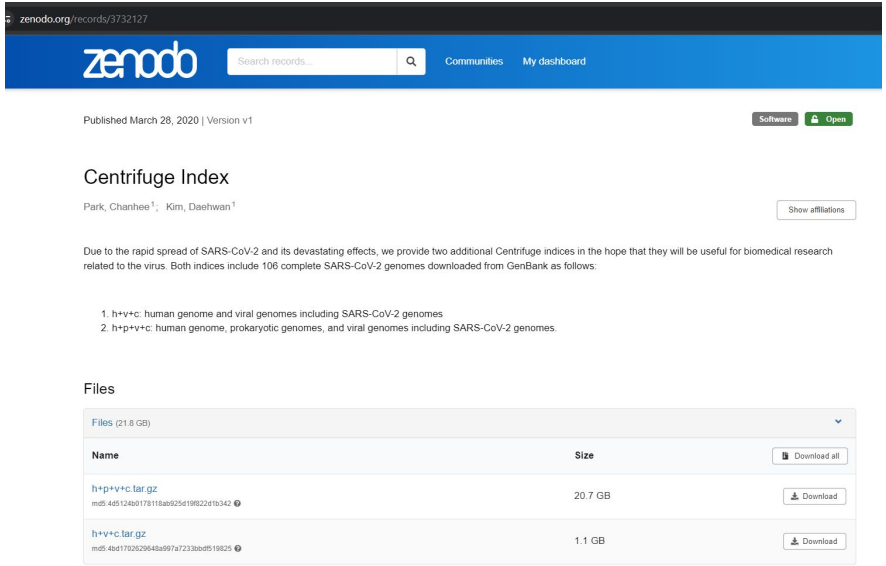
```
--2024-03-05 22:21:39--  https://zenodo.org/records/3732127/files/h+p+v+c.tar.gz
Resolving zenodo.org (zenodo.org)... 188.184.103.159, 188.184.98.238, 188.185.79.172, ...
Connecting to zenodo.org (zenodo.org)|188.184.103.159|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20737377933 (19G) [application/octet-stream]
Saving to: ‘h+p+v+c.tar.gz’

h+p+v+c.tar.gz      100%[===================>]  19.31G  13.1MB/s    in 26m 6s

2024-03-05 22:47:47 (12.6 MB/s) - ‘h+p+v+c.tar.gz’ saved [20737377933/20737377933]
```

[7]:
```
!tar -xvzf h+p+v+c.tar.gz
```

```
hpvc.1.cf
hpvc.2.cf
hpvc.3.cf
hpvc.4.cf
```



Centrifuge Command Parameters

Index Selection

-x <cf-idx>: Index filename prefix (minus trailing .X.cf)

Input Options

-1 <m1> -2 <m2>: Paired-end input files

-U <r>: Unpaired reads

Output Configuration

-S <filename>: File for classification output (stdout by default)

--report-file <report>: Tabular report output file (centrifuge_report.tsv by default)

Classification Settings

--host-taxids <taxids>: Preferred taxonomic IDs in classification

--exclude-taxids <taxids>: Excluded taxonomic IDs in classification

Performance

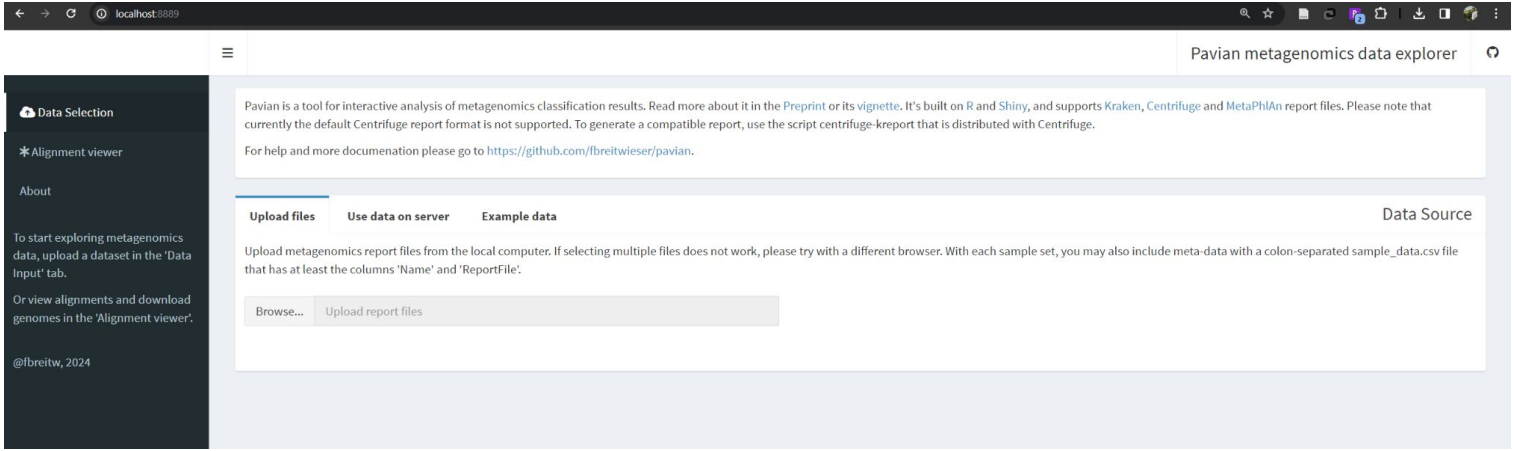
-p/--threads <int>: Number of alignment threads (default: 1)

--mm: Use memory-mapped I/O for index; suitable for multiple instances

```
!centrifuge -x hpvc -U mappings/evol1.sorted.unmapped.R1.fastq --report-file report1.txt -S results1.txt
```

Pavian : Interactive Analysis of Metagenomics Data

Pavian is a interactive browser application for analyzing and visualization metagenomics classification results from classifiers such as Kraken, Kraken_Uniq, Kraken 2, Centrifuge and MetaPhlAn.



Pavian : Interactive Analysis of Metagenomics Data

1

```
!centrifuge-kreport -x hpvc  results.txt > read_classifications.tsv.kraken
```

```
Loading taxonomy ...
Loading names file ...
/home/sr3622/miniconda3/bin/centrifuge-inspect:24: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see the module's documentation for alternative uses
  import imp
Loading nodes file ...
/home/sr3622/miniconda3/bin/centrifuge-inspect:24: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see the module's documentation for alternative uses
  import imp
```

2

```
import ipywidgets as widgets
from epi2melabs.notebook import InputForm, InputSpec

pavian_form = InputForm(
    InputSpec('port', 'Aux. EPI2ME Labs port', widgets.IntText(8889)))
pavian_form.display()
```

```
VBox(children=(HBox(children=(Label(value='Aux. EPI2ME Labs port', layout=Layout(width='150px')), interactive(…
```

```
# running Pavian web server
# !echo "Checking pavian install..."

script = """
ncpus=parallel::detectCores()
options(Ncpus=ncpus)
remotes::install_github("fbreitwieser/pavian", upgrade=T, quiet=T)"""
_script = os.path.expanduser("~/.pavian_install.R")
with open(_script, "w") as fh:
    fh.write(script)
!Rscript $_script
!echo "Done."
!echo "Running pavian..."
port = pavian_form.port
!R -e "pavian::runApp(host='0.0.0.0', port="$port")"
```

Open Url : localhost:8889

```
Done.
Running pavian...
```

Centrifuge Output

Classification Output

```
readID  seqID   taxID   score   2ndBestScore    hitLength   queryLength numMatches
NS500207:12:H04WYAFXX:3:21408:22104:9367    NZ_CP014768.1   1813821 81  81  24  27  2
NS500207:12:H04WYAFXX:3:21408:22104:9367    species 562 81  81  24  27  2
NS500207:12:H04WYAFXX:4:21410:2455:20204    unclassified    0   0   0   0   16  1
NS500207:12:H04WYAFXX:2:21105:21680:16668   unclassified    0   0   0   0   21  1
NS500207:12:H04WYAFXX:3:11609:20259:9826    unclassified    0   0   0   0   15  1
```

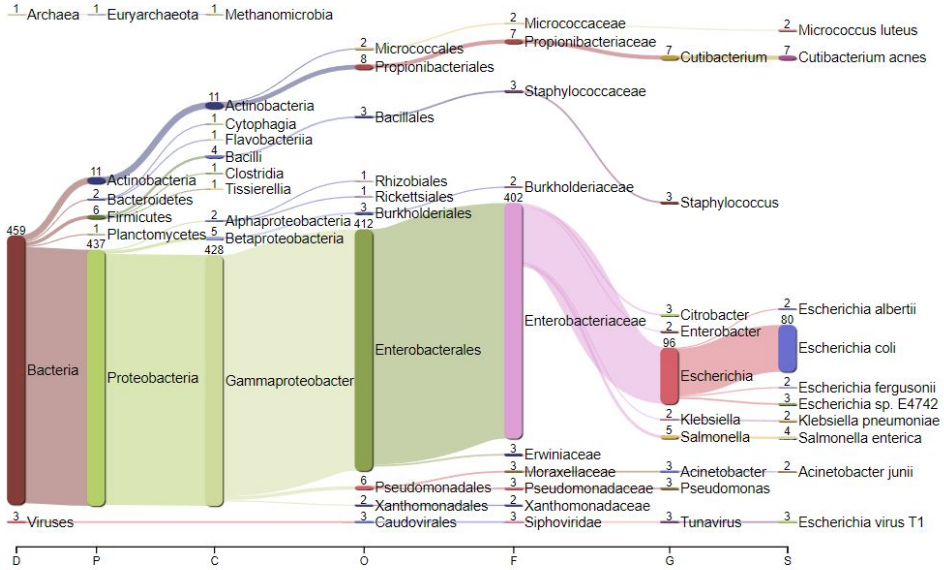
Report output

```
name    taxID   taxRank genomeSize  numReads    numUniqueReads  abundance
Cellvibrio  10  genus   4576573 1   0   0.0
Pseudomonas fluorescens 294 species 6604895 2   0   0.0
Pseudomonas oleovorans  301 species 4686340 1   1   0.0
Xanthomonas 338 genus   23419814    1   0   0.0
Neisseria   482 genus   2223758 1   0   0.0
```

Pavian: Sankey visualization

Let see the HTML output!

read_classifications.tsv.kraken



Resources

https://labs.epi2me.io/notebooks/Metagenomic_classification_tutorial.html?

Kim, D., Song, L., Breitwieser, F. P., & Salzberg, S. L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome research*, *26*(12), 1721-1729.

Breitwieser, F. P., & Salzberg, S. L. (2020). Pavian: interactive analysis of metagenomics data for microbiome studies and pathogen identification. Bioinformatics, 36(4), 1303-1304.