3000788 Intro to Comp Molec Biol

Week 6: RNA sequencing analysis demo

Fall 2024

Sira Sriswasdi, PhD

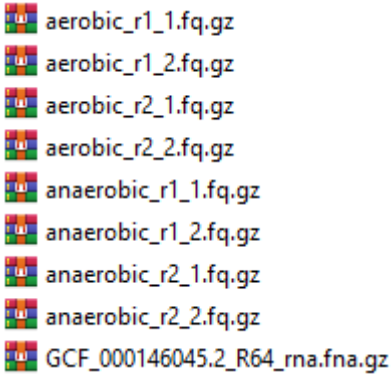
- Research Affairs
- Center of Excellence in Computational Molecular Biology (CMB)
- Center for Artificial Intelligence in Medicine (CU-AIM)

Today's activity

- Check quality of FASTQ with FastQC
- Process RNA-seq data from FASTQ using *kallisto*
- Perform differential expression analysis in *sleuth* R package
- Perform functional enrichment analysis on *WebGestalt*
- https://www.webgestalt.org/

Preparations

- Get the FASTQ data and yeast reference transcriptome
- https://figshare.com/articles/dataset/Yeast_RNA-seq_data_and_transcriptome_for_kallisto-sleuth_demo_session/24182520

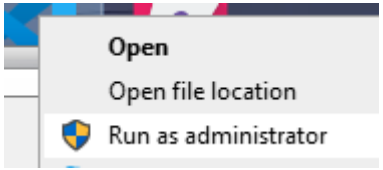


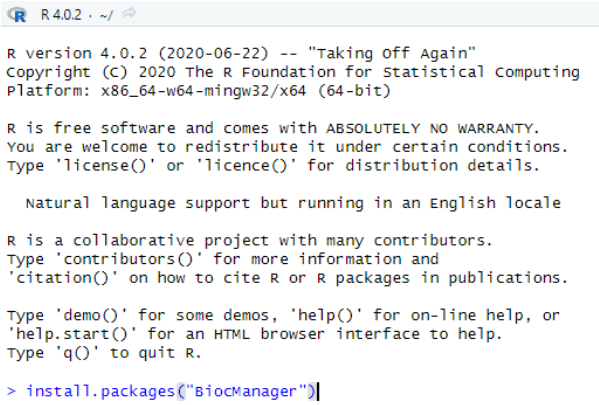
- Install *R* & *RStudio*
- Install *sleuth* (see next slide)
- Get *run_sleuth.R* from course website's *demo* folder

- Download *kallisto* (version 0.46.1, select your OS)
- https://pachterlab.github.io/kallisto/download

Preparations

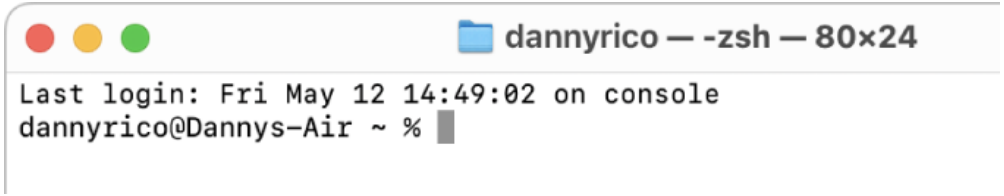
- Install *sleuth*
  - You may have to run RStudio as administration
  - In the command area, type
    - install.packages("BiocManager")
    - BiocManager::install("rhdf5")
    - install.packages("devtools")
    - devtools::install_github("pachterlab/sleuth")
  - To test that you were successful, type
    - library(sleuth)
  - If the program asks: Update all/some/none? [a/s/n], it is safe to choose “n”.

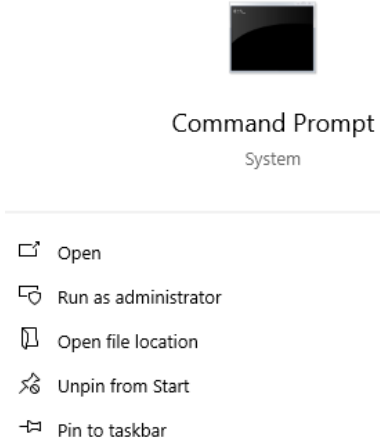




Command line interface

Terminal and Command Prompt






```
C:\Users\Sira\Downloads\yeast_data>kallisto index -i yeast_rna GCF_000146045.2_R64_rna.fna.gz

[build] loading fasta file GCF_000146045.2_R64_rna.fna.gz
[build] k-mer length: 31
[build] counting k-mers ... done.
[build] building target de Bruijn graph ...  done
[build] creating equivalence classes ...  done
[build] target de Bruijn graph has 11192 contigs and contains 8200305 k-mers
```

How to run script in command line?

```
kallisto 0.50.0

Usage: kallisto <CMD> [arguments] ..

Where <CMD> can be one of:

    index         Builds a kallisto index
    quant         Runs the quantification algorithm
    quant-tcc     Runs quantification on transcript-compatibility counts
    bus           Generate BUS files for single-cell data
    h5dump        Converts HDF5-formatted results to plaintext
    inspect       Inspects and gives information about an index
    version       Prints version information
    cite          Prints citation information

Running kallisto <CMD> without arguments prints usage information for <CMD>
```

Let’s index the yeast reference transcriptome

```
kallisto 0.50.0
Builds a kallisto index

Usage: kallisto index [arguments] FASTA-files

Required argument:
-i, --index=STRING          Filename for the kallisto index to be constructed

Optional argument:
-k, --kmer-size=INT         k-mer (odd) length (default: 31, max value: 31)
-d, --d-list=STRING         Path to a FASTA-file containing sequences to mask from quantification
    --make-unique           Replace repeated target names with unique names
    --aa                    Generate index from a FASTA-file containing amino acid sequences
    --distinguish           Generate index where sequences are distinguished by the sequence name
-t, --threads=INT           Number of threads to use (default: 1)
```

kallisto index -i yeast_transcriptome GCF_000146045.2_R64_rna.tna.gz

kallisto quant

```
Usage: kallisto quant [arguments] FASTQ-files

Required arguments:
-i, --index=STRING            Filename for the kallisto index to be used for
                              quantification
-o, --output-dir=STRING       Directory to write output to

Optional arguments:
-b, --bootstrap-samples=INT   Number of bootstrap samples (default: 0)
    --seed=INT                Seed for the bootstrap sampling (default: 42)
    --plaintext               Output plaintext instead of HDF5
    --single                  Quantify single-end reads
    --single-overhang         Include reads where unobserved rest of fragment is
                              predicted to lie outside a transcript
    --fr-stranded             Strand specific reads, first read forward
    --rf-stranded             Strand specific reads, first read reverse
-l, --fragment-length=DOUBLE  Estimated average fragment length
-s, --sd=DOUBLE               Estimated standard deviation of fragment length
                              (default: -l, -s values are estimated from paired
                               end data, but are required when using --single)
-t, --threads=INT             Number of threads to use (default: 1)
```

One output folder per sample

Bootstrap estimates technical variances

More CPUs = faster

Differential expression

Output from kallisto quant

- Four output folders (= number of samples)

- Look for abundance.h5 and abundance.tsv
  - abundance.h5 contains bootstrapping results

Building a metadata table for *sleuth*

- Two required columns: sample and path
  - sample = name of each sample
  - path = directory to *kallisto* output

- condition will be used as an experimental design factor

	A	B	C
1	sample	condition	path
2	aerobic1	aerobic	C:\Users\Sira\Downloads\yeast_data\aerobic1
3	aerobic2	aerobic	C:\Users\Sira\Downloads\yeast_data\aerobic2
4	anaerobic1	anaerobic	C:\Users\Sira\Downloads\yeast_data\anaerobic1
5	anaerobic2	anaerobic	C:\Users\Sira\Downloads\yeast_data\anaerobic2

Editing and running R script

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

run_sleuth.R

Source on Save Run Source

```r
## Load sleuth library
library('sleuth')

## Set working directory (so that files will be read from/written to this location)
setwd('C:\\Users\\Sira\\Downloads\\yeast_data')

## Load metadata table
s2c <- read.table(file.path('metadata.txt'), header = TRUE, stringsAsFactors=FALSE)

## Preprocess data into sleuth format
#### We use bootstrapping data from kallisto here
so <- sleuth_prep(s2c, extra_bootstrap_summary = TRUE, read_bootstrap_tpm = TRUE)

## Fitting of the alternative hypothesis = condition-specific expression
so <- sleuth_fit(so, ~condition, 'full')

## Fitting of the null hypothesis = no difference across condition
so <- sleuth_fit(so, ~1, 'reduced')
```

Outputs from *sleuth*

- Table of transcript and p-value (and q-value)

A	B	C	D	E	F	G	H	I
	target_id	pval	qval	test_stat	rss	degrees_free	mean_obs	var_obs
1	NM_001181124.1	2.59E-10	1.27E-06	39.96539563	15.36339695	1	8.417310722	5.121132316
2	NR_132186.1	6.55E-10	1.27E-06	38.14979375	3.640905722	1	9.879766586	1.213635241
3	NR_132187.1	6.55E-10	1.27E-06	38.14979375	3.640905722	1	9.879766586	1.213635241
4	NM_001178902.1	4.30E-09	1.45E-06	34.48167964	6.678866411	1	8.185847672	2.226288804
5	NM_001179305.1	6.06E-09	1.45E-06	33.81376634	0.431345958	1	11.06920134	0.143781986
6	NM_001179347.3	6.59E-09	1.45E-06	33.65244067	14.28124796	1	8.201656196	4.760415988
7	NM_001180385.3	5.58E-09	1.45E-06	33.97442784	1.862161499	1	9.003896352	0.6207205
8	NM_001180810.3	6.39E-09	1.45E-06	33.71281635	2.818466062	1	8.519619279	0.939488687

- Table of expression level (in TPM)

A	B	C	D	E
	aerobic1	aerobic2	anaerobic1	anaerobic2
NM_001178148.1	9.924504945	9.865041182	10.84427431	9.033759097
NM_001178149.1	9.138502413	8.571050881	9.582161084	10.74528304
NM_001178150.1	1171.672132	1170.681103	2148.017091	2154.089911
NM_001178151.1	190.2430374	196.6867042	89.27467731	91.84976857
NM_001178152.1	167.4525834	167.4358872	120.3110809	118.6026502
NM_001178153.1	31.80449311	40.69292227	28.71252687	33.11786185

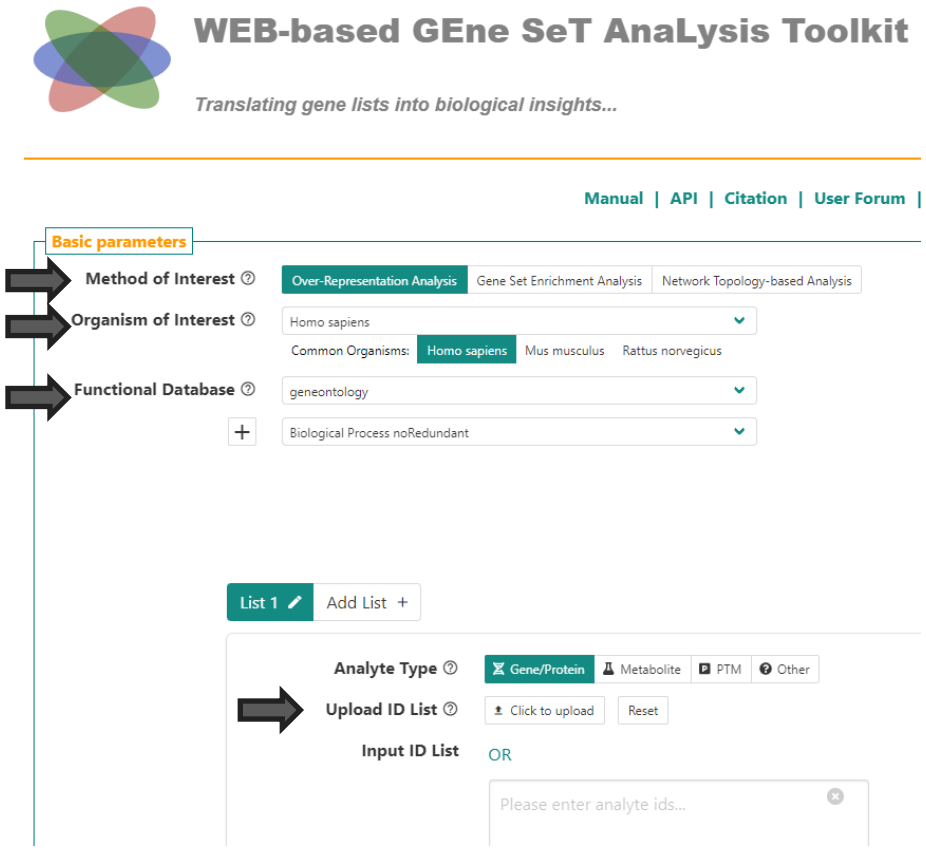
Functional enrichment

WebGestalt

- Provide all 3 methods
  - Overrepresentation
  - GSEA
  - Network-based

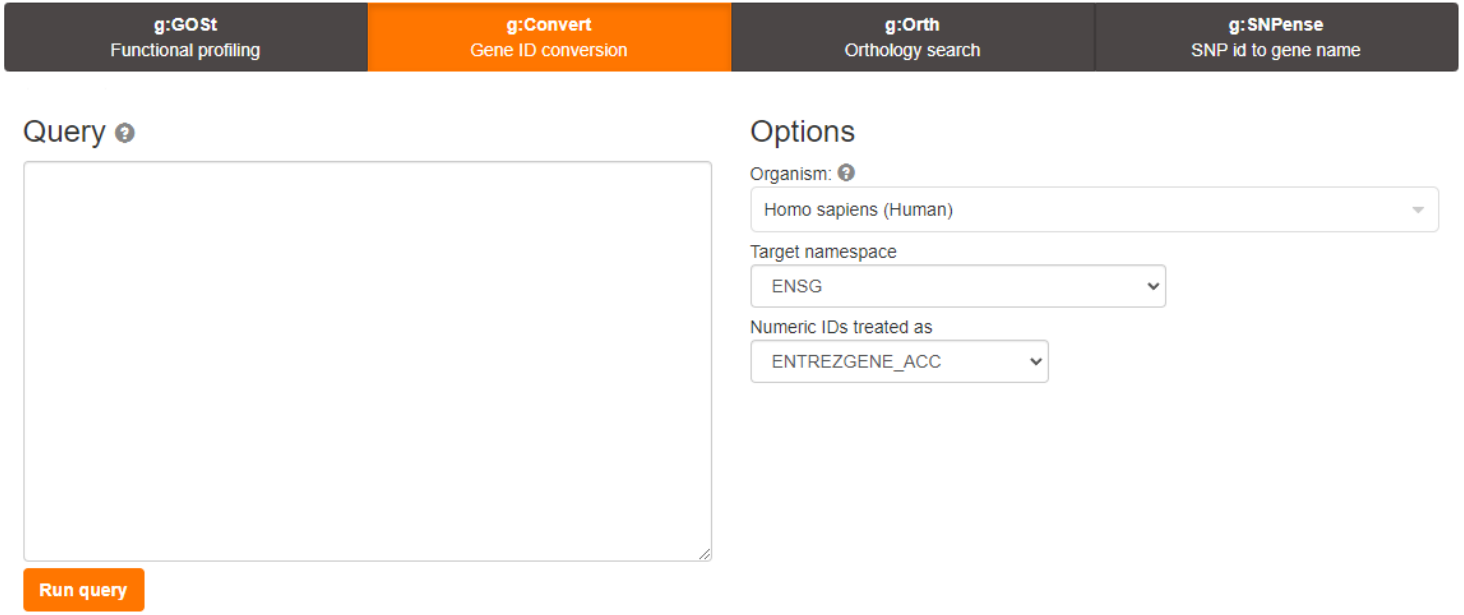
- Choices of organisms and functional databases

- Recognize some ID formats



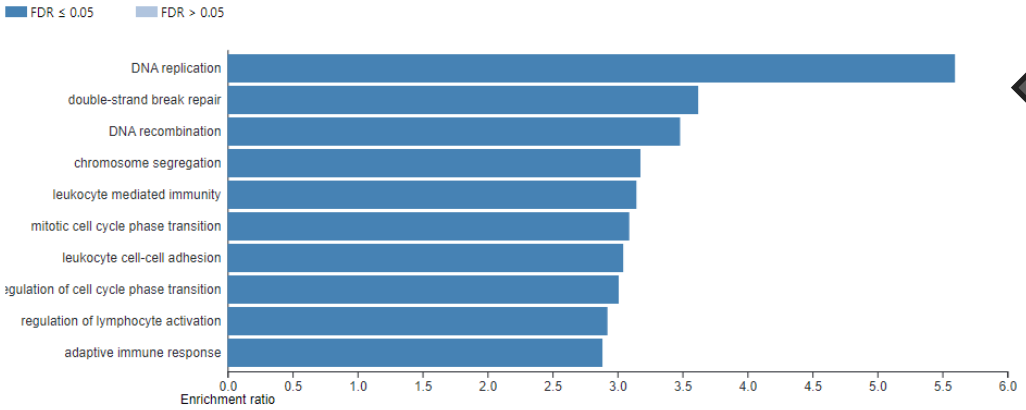
g:Profiler

Gene/transcript ID converter

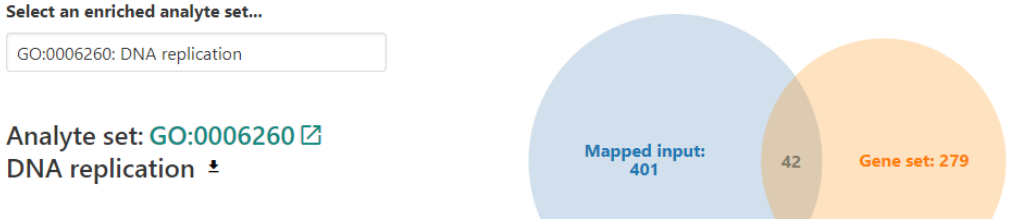


https://biit.cs.ut.ee/gprofiler/convert

WebGestalt output



Enriched pathways or biological functions



Number of genes involved

Any question?