RNA-Seq Analysis

Author: Tony Kabilan Okeke
Date: 22 May 2022

In this assignment, we you will re-analyze the RNA-Seq data reported in the following paper: *Regulation of Glucose-Dependent Expression by the RNA Helicase Dbp2 in Saccharomyces cerevisiae*

The corresponding data for this study is at GSE58097.

The following runs were included in this analysis:

- Wild Type - SRR1302790
- Mutant - SRR1302792

In [ ]:

```
%load_ext autoreload
%autoreload 2

# Imports
from urllib.request import urlretrieve
from multiprocessing import cpu_count
from glob import glob
import pandas as pd
import numpy as np
import subprocess
import os

# Paths to command line tools
FEATURECOUNTS = "/home/kabil/.anaconda3/envs/binf/bin/featureCounts"
FASTQDUMP = "/home/kabil/.anaconda3/envs/binf/bin/fastq-dump"
PREFETCH = "/home/kabil/.anaconda3/envs/binf/bin/prefetch"
BWA = "/home/kabil/.anaconda3/envs/binf/bin/bwa"

# Directories and files
fastqdir = "/mnt/h/data/fastq"
samdir = "/mnt/h/data/samfiles"
yeastgenome = "/mnt/h/data/refseq/NC_001133.9_genomic.fna.gz"
yeastannot = "/mnt/h/data/refseq/NC_001133.9_genomic.gtf.gz"
genomelink = "https://ftp.ncbi.nih.gov/genomes/refseq/fungi/Saccharomyces_cerevisiae/reference/GCF_000146045.
annotlink = "https://ftp.ncbi.nih.gov/genomes/refseq/fungi/Saccharomyces_cerevisiae/reference/GCF_000146045.2
```

Download `fastq` Files from SRA

In [ ]:

```
# Load fastq files
runs = ['SRR1302790', 'SRR1302792']

for run in runs:
    if not os.path.exists(f"{fastqdir}/{run}_pass_1.fastq.gz"):
        # This produces 2 (paired end) files for each read
        cmd = (f"{FASTQDUMP} --outdir {fastqdir} --skip-technical --gzip " +
               f"--read-filter pass --dumpbase --split-3 --clip {run}")

        subprocess.call(f"{PREFETCH} {run}", shell=True)
        subprocess.call(cmd, shell=True)
    else:
        print(f"{run} fastq files already exist.")
```

```
SRR1302790 fastq files already exist.
SRR1302792 fastq files already exist.
```

Download and Index the Yeast Genome

The yeast genome was retrieved through the NCBI FTP site at /genomes/refseq/Saccharomyces_cerevisiae/.../GCF_000146045.2_R64_genomic.fna.gz. The genome annotation file was also downloaded from the same site.

In [ ]:

```
# Download genome
if not os.path.exists(yeastgenome):
    print("Downloading yeast reference genome...")
    urlretrieve(genomelink, yeastgenome)

# Download annotation
if not os.path.exists(yeastannot):
    print("Downloading genome annotation...")
    urlretrieve(annotlink, yeastannot)

# Index genome
if not os.path.exists(yeastgenome + '.bwt'):
    print("Indexing yeast reference genome...")
    cmd = f"{BWA} index '{yeastgenome}'"
    subprocess.call(cmd, shell=True)
```

Map Reads to Yeast Reference Genome

Running `fastq-dump` with the `--split-3` option produces 2 files for each run; for each run, both files will be passed to `bwa` as a read pair, so only 2 SAM files will be created.

In [ ]:

```
# Create read pairs
readpairs = []
for run in runs:
    readpairs.append( ' '.join(glob(f"{fastqdir}/{run}_pass_[12].fastq.gz")) )

# Map reads to the reference genome
samfiles = []
for run, readpair in zip(runs, readpairs):
    samfile = f"{samdir}/{run}.sam"

    if not os.path.exists(samfile):
        cmd = f"{BWA} mem -t {cpu_count()} {yeastgenome} {readpair} > {samfile}"
        subprocess.call(cmd, shell=True)
    else:
        print(f"{run} has already been mapped.")

    samfiles.append(samfile)
```

```
SRR1302790 has already been mapped.
SRR1302792 has already been mapped.
```

Feature Counts

In [ ]:

```
# Generate table of feature counts for WT and Mutant samples
if not os.path.exists("results/feature_counts.txt"):
    cmd = (f"{FEATURECOUNTS} -p -a {yeastannot} -o results/feature_counts.txt " +
           f"{' '.join(samfiles)} -O -T {cpu_count()} " +
           f"--tmpDir /mnt/h/tmp/ -t exon -g gene_id")
    subprocess.call(cmd, shell=True)
else:
    print('feauture_counts.txt already exists.')
```

```
feauture_counts.txt already exists.
```

In [ ]:

```
# Load feature counts
df = pd.read_table('results/feature_counts.txt', skiprows=1) \
    .rename({f'{samdir}/SRR1302790.sam': 'WT',
             f'{samdir}/SRR1302792.sam': 'Mutant',
             'Geneid': 'Gene'}, axis=1)
# Use pseudo-counts to avoid inf fold changes
df[['WT', 'Mutant']] = df[['WT', 'Mutant']] + 1

# Perform TPM Normalization
df[['WT', 'Mutant']] = df[['WT', 'Mutant']].div(df.Length, axis=0)
df[['WT', 'Mutant']] = df[['WT', 'Mutant']] / df[['WT', 'Mutant']].sum() * 1e6
```

Differential Gene Expression

In [ ]:

```
# Compute fold-change between mutant and WT
fc = df['Mutant'] / df['WT']
df['FoldChange'] = np.where(fc < 1, -1/fc, fc)  # signed fold-change

# List 10 most different denes between groups
df = df.sort_values('FoldChange', key=abs, ascending=False) \
    .reset_index(drop=True)
df[['Gene', 'Chr', 'FoldChange']].head(10)
```

Out[ ]:

	Gene	Chr	FoldChange
0	YFL014W	NC_001138.5	801.449474
1	YBR115C	NC_001134.8	-323.410199
2	YPR157W	NC_001148.4	321.143255
3	YDL048C	NC_001136.10	299.833276
4	YGR248W	NC_001139.9	277.168362
5	YGR052W	NC_001139.9	276.768792
6	YNL112W	NC_001146.8;NC_001146.8	-264.086003
7	YBR054W	NC_001134.8	247.015566
8	YLR297W	NC_001144.5	243.609418
9	YGR138C	NC_001139.9	234.155755

Functional Enrichment Analysis

Only genes that exhibit a 5-fold change or higher were included in the functional enrichment analysis.

The functional enrichment results are stored in the `results/enrichment_go_terms.txt` and `results/enrichment_kegg_paths.txt` ; the top 10 enriched go terms and kegg pathways are shown below.

In [ ]:

```
fc_thr = 5
genes = df[df['FoldChange'] > fc_thr].Gene.to_list()

# Display DAVID results
for file in ['go_terms', 'kegg_paths']:
    tbl = pd.read_table(f'results/enrichment_{file}.txt')
    display(
        tbl.loc[:10, ['Term', 'PValue', 'Count']].style \
            .set_caption('Enriched ' + file.replace('_', ' ').upper())
    )
```

Enriched GO TERMS

	Term	PValue	Count
0	GO:0031505~fungal-type cell wall organization	0.000000	53
1	GO:1902600~hydrogen ion transmembrane transport	0.000000	38
2	GO:0055085~transmembrane transport	0.000000	73
3	GO:1904659~glucose transmembrane transport	0.000000	15
4	GO:0015761~mannose transport	0.000000	14
5	GO:0015755~fructose transport	0.000000	14
6	GO:0008645~hexose transport	0.000000	14
7	GO:0008643~carbohydrate transport	0.000000	19
8	GO:0030435~sporulation resulting in formation of a cellular spore	0.000001	36
9	GO:0006122~mitochondrial electron transport, ubiquinol to cytochrome c	0.000003	11
10	GO:0006754~ATP biosynthetic process	0.000035	10

Enriched KEGG PATHS

	Term	PValue	Count
0	sce01100:Metabolic pathways	0.000000	142
1	sce00190:Oxidative phosphorylation	0.000000	34
2	sce01200:Carbon metabolism	0.000031	28
3	sce00500:Starch and sucrose metabolism	0.000052	15
4	sce00010:Glycolysis / Gluconeogenesis	0.001567	15
5	sce04113:Meiosis - yeast	0.001569	27
6	sce01110:Biosynthesis of secondary metabolites	0.002828	55
7	sce00520:Amino sugar and nucleotide sugar metabolism	0.008197	10
8	sce01250:Biosynthesis of nucleotide sugars	0.008618	8
9	sce00730:Thiamine metabolism	0.009211	7
10	sce00052:Galactose metabolism	0.011068	8