MicroRNA Analysis

Author: Tony Kabilan Okeke

In [ ]:
```
%load_ext autoreload
```

In [ ]:
```
# Imports
%autoreload 2
import os
import bmes
import rich
import sqlite3
import numpy as np
import pandas as pd
from scipy.stats import ttest_ind
from IPython.display import display, HTML
from statsmodels.stats.multitest import fdrcorrection

os.chdir("/home/kabil/tko35/bmes543/code/mirna")
```

In [ ]:
```
# Definitions
def targetscandb_mir2target(mirna: str, scorethr: float=0.8) -> list:
    """
    Code was originally written by Dr. Ahmet Sacan <ahmetmsacan@gmail.com>
    """

    # Download and connect to database
    dbfile = bmes.downloadurl('http://sacan.biomed.drexel.edu/ftp/binf/targetscandb.sqlite')
    conn = sqlite3.connect(dbfile)
    cur = conn.cursor()

    # Construct query
    query = f"""SELECT distinct("generefseqid") FROM "mir2target"
                WHERE score>={scorethr:f}
                AND mirna IN ("{mirna}","{mirna}-3p","{mirna}-5p")"""

    # Query database and return results
    cur.execute(query)
    rows = cur.fetchall()
    return [ row[0] for row in rows ]
```

Load Unfiltered miRNA Dataset

In [ ]:
```
# Load Unfiltered CRPS Data  (replace Infs with NaNs)
unfilt = pd.read_excel("CRPS_unfiltered.xlsx", header=None,
                       skiprows=1, index_col=0) \
           .rename_axis('miRNA') \
           .replace({np.inf: np.nan})
# Set column names
cols = pd.read_excel("CRPS_unfiltered.xlsx", header=None, nrows=1).values[0]
unfilt.columns = cols[1:]
```

Remove miRNA Detected in 3 or Fewer Samples

In [ ]:
```
filt = unfilt[ (~unfilt.isna()).sum(axis=1) > 3 ].copy()
```

Replace Undetected Values (`Inf`) with the Average Expression of the miRNA in the Remaining Samples

In [ ]:
```
filt = filt.apply(lambda row: row.fillna(row.mean()), axis=1)
```

Show First 5 Genes for the First 6 Samples

In [ ]:
```
rich.print( filt.iloc[:5,:6].round(4) )
```

miRNA	control	control	control	control	control	control
hsa-miR-425	18.4318	18.5392	19.2871	17.9853	19.9477	20.3738
hsa-miR-129-5p	33.9620	33.6747	36.4027	33.9620	33.9620	33.9620
hsa-miR-329	33.5697	31.3315	34.2243	29.5776	37.7981	32.5906
hsa-miR-484	14.2823	14.2317	14.2522	13.9903	14.6762	14.3608
hsa-miR-625	25.4440	25.1881	25.4555	24.2658	27.1671	25.5140

Compute ΔCT Values

Use *RNU44*, *RNU48* and *MammU6* as endogenous controls for calculating CT0.

In [ ]:
```
# Comute CT0 valyes
CT0 = filt[ filt.index.isin(['RNU44', 'RNU48', 'MammU6']) ].mean(axis=0)

# Subtract Sample CT0 Values from CT Values
filt = filt.sub(CT0, axis=1)
```

Compute $\Delta\Delta CT$ Values

In [ ]:
```
filt['deltadeltaCT'] = filt['patient'].mean(axis=1) - filt['control'].mean(axis=1)
```

Compute Fold Changes

In [ ]:
```
# Compute fold change
filt['FC'] = 2 ** -filt['deltadeltaCT']

# Replace values < 1 with their negative inverse (Signed Fold Change)
filt['FC'] = np.where(filt['FC'] < 1, -1/filt['FC'], filt['FC'])
```

Show the Top 10 Most Changing miRNAs

In [ ]:
```
rich.print(filt['FC'].sort_values(key=abs, ascending=False).head(10))
```

```
miRNA
hsa-miR-939    -5.757791
hsa-miR-25#    -5.077934
hsa-miR-17#    -4.282382
hsa-miR-223    -3.894710
hsa-miR-29b    -3.580711
hsa-let-7c     -3.229430
hsa-miR-133b   -3.139277
hsa-miR-18b    -3.114760
hsa-let-7b     -3.089840
hsa-miR-190    -3.059572
Name: FC, dtype: float64
```

Find Significantly Different miRNAs (Controls vs Patients)

In [ ]:
```
# Compute p-values
filt['p-value'] = ttest_ind(
    filt['control'], filt['patient'], axis=1
).pvalue
# FDR Correction for p-values (q-values)
filt['q-value'] = fdrcorrection(filt['p-value'])[1]

# Print the Top 10 Most Significantly Different Genes
rich.print(filt['q-value'].sort_values(ascending=True).head(10))
```

```
miRNA
hsa-miR-320B      0.000126
hsa-let-7b        0.000126
hsa-miR-25#       0.000126
hsa-let-7c        0.000126
hsa-miR-320       0.000126
hsa-miR-939       0.000134
hsa-miR-629       0.000343
hsa-let-7d        0.001145
hsa-miR-132       0.001145
hsa-miR-532-3p    0.001145
Name: q-value, dtype: float64
```

Find Which mRNAs are the Predicted Targets of the Significant miRNAs from the CRPS Study Using `TargetScan`

In [ ]:
```
# Select significantly different miRNAs
# q-value threshold:  < .001
# fold change threshold:  > |3|
I = (filt['FC'].abs() >= 3) & (filt['q-value'] <= 0.01)
miRNAs = filt[I].index.to_list()

# Find which mRNAs are Predicted Targets of the Significant miRNAs
targets = [targetscandb_mir2target(mirna, .95) for mirna in miRNAs]
targets = np.unique([ target for sub in targets for target in sub ])
```

Target Enrichment

The results enriched pathways and gene ontology biological process terms returned by DAVID are stored in the `Enriched_GOBP.txt` and `Enriched_KEGG.txt` files respectively.

In [ ]:
```
# Display the top 3 most significantly enriched pathways and GO BPs
for file in ['KEGG_Pathway', 'GO_BP']:
    tbl = pd.read_table(f'Enriched_{file}.txt')
    display(
        tbl.loc[:2, ['Term', 'PValue', 'Count']].style \
            .set_caption('Enriched ' + file.replace('_', ' '))
    )
```

Enriched KEGG Pathway

	Term	PValue	Count
0	hsa04150:mTOR signaling pathway	0.000214	10
1	hsa04550:Signaling pathways regulating pluripotency of stem cells	0.000597	9
2	hsa04152:AMPK signaling pathway	0.001034	8

Enriched GO BP

	Term	PValue	Count
0	GO:0017148~negative regulation of translation	0.000627	7
1	GO:0071363~cellular response to growth factor stimulus	0.000805	6
2	GO:0032924~activin receptor signaling pathway	0.001642	4