Graph Wavelet Based Denoising Gene Expression Data Python Code
"# NetWave"
One needs to have python 3.0+ installed in their machine. Following are the dependencies of the code :
Numpy, Pandas, Pygsp, scipy, sklearn
R dependencies :
minet,methods,GENIE3,propr
You have to download python code in your local machine/server.
For execution you have to pass filename of expression csv as a input file.
Expression csv file should not contain header and genes i.e. it consist of only data on which filtering is going to perf
orm.
Row represents samples and column represent genes in csv file.
Single cell data should be fpkm.
USE THE FOLLOWING COMMANDS :
```bash
python3 NetWave.py -e -f -t -k -p -o
e.g.
python3 NetWave.py -e net1.txt -f wavelet -t bulk -k 40 -p 70 -o net1_output.txt
```
'net1.txt' consist of expression data to be filtered.
Regarding each iput parameter :
-e : Path to expression data
-o : Path to save filterd expression
-f : Filter method to use (Options are : chebychev, wavelet, SureShrink, BayesShrink, Gaussian_MDL_hardthresh)
-t : data type (Options : single_cell / bulk)
-k : KNN value (min(genes,samples) in expression data)
-p : Percentile Frequency cutoff
Differential Centrality Python Code
"# centrality.py"
One needs to have python 3.0+ installed in their machine. Following are the dependencies of the code :
Numpy, Pandas, Pygsp, scipy, sklearn, networkx
R dependencies : minet,methods,GENIE3,propr
You have to download python code in your local machine/server.
For execution you have to pass filename of young and old filtered expression csv as a input file.
Expression csv file should not contain header and genes i.e. it consist of only data on which filtering is going to perf
orm.
Row represents samples and column represent genes in csv file.
User gets :::
1. Differential degree and pagerank for old and young.
2. centrality (degree, pagerank, betweenness, closeness & eigen values) for both young and old in separate files by t
he default names "young_centrality.txt" and "old_centrality.txt"
USE THE FOLLOWING COMMANDS :
```bash
python3 centrality.py -y -o -g -c -r -t
e.g.
python3 centrality.py -y exp_young.txt -o exp_old.txt -g genes.txt -c pearson -r diff_output.txt -t 100000
```
'exp_young.txt' consist of filtered young expression data, 'exp_old.txt' consist of filtered old expression data.
Regarding each iput parameter :
-y : Path to young filtered expression data
-o : Path to Old filtered expression data
-g : Path to Gene list
-r : Path to save differential centrality result

-c : Options : pearson, spearman, aracne, genie3, phi, rho
-t : Number of Top Edges to select to build network

Overlap with goldenset Python Code
"# NetWave Python"

One needs to have python 3.0+ installed in their machine. Following are the dependencies of the code :
Numpy, Pandas, Pygsp, scipy, sklearn
R dependencies :
minet,methods,GENIE3,propr

You have to download python code in your local machine/server.
For execution you have to pass filename of expression csv as a input file.
Expression csv file does not contain header and genes i.e. it consist of only data on which filtering is going to perfor
m.
Row represents samples and column reprent genes in csv file.

USE THE FOLLOWING COMMANDS :
```bash
python3 NetWave.py -e <exp_path> -g <goldset_path> -f <filter_method> -c <correlation_method> -t <data_type>
-k <KNN> -d <dx> -p <cutoff(wavelet)>

e.g.
python3 NetWave.py -e net1.txt -g goldset_net1.txt -f wavelet -c pearson -t bulk -k 40 -d 10000 -p 70
```
'net1.txt' consist of expression data to be filtered.

Regarding each iput parameter :
-e <expression Data> : Path to expression data
-g <golden set> : Path to golden set
-f : Filter method to use (Options are : chebychev, wavelet, SureShrink, BayesShrink, Gaussian_MDL_hardthresh)
-c : correlation method to use (Options : pearson, spearman, aracne, genie3, phi, rho)
-t : data type (Options : single_cell / bulk)
-k : KNN value (<min(genes,samples) in expression data)
-d : dx for AUC
-p : Percentile Frequency cutoff