preprocessing

October 11, 2019

0.1 Preprocessing

- Join datasets with the GPL file to populate the mRNA_Accession on principal file.
- The identifier is 'ID'

0.2 Importing packages

In [1]:
```
#============================================================
# Packages
#============================================================

import pandas as pd
import numpy as np
import os # Accesing to directory
import re # Regular Expressions
from six.moves import reduce # Merge dataframes


## Setting the seed value for reproducibility

seed_value= 123# Set a seed value

# Set `python` built-in pseudo-random generator at a fixed value
import random
random.seed(seed_value)

# Set `numpy` pseudo-random generator at a fixed value
import numpy as np
np.random.seed(seed_value)

seed =  np.random.RandomState(123)
# do not call numpy.random.rand() but seed.rand()


# 3. Set environment
os.urandom(seed_value)
```

Out[1]:
```
b'e\x8eu)#\xf3\x98\xac\xef\xeb\xb0\xac\x7f\xc9(Y\x1c\xe3\xe9\x9d\xeci\xd3P\xb2V`&\xb4\x
```

0.3 Defining Functions

- Defining function to be used in this script

In [2]:
```
#============================================================
# Reading Files
#============================================================

def read_files(file, file_GPL, skip_rows =19,
               columns_to_keep =  ['ID', 'mrna_assignment']):
    '''
    Function that reads two files as dataframes, the principal dataset and the
    GPL files with the mrna_assignment values
    transform 'ID' to object data type.
    Input: User need to provide the path for both files, where path1 is the
    principal file.
    The number of rows to skip for the second file at the beginning of the dataset
    and the columns names.
    Output = two dataframes
    '''
    df_file = pd.read_csv(file, delimiter="\t", dtype = {'ID': str})
    df_file_GPL = pd.read_csv(file_GPL , delimiter="\t" , skiprows= skip_rows,
                              usecols= columns_to_keep, dtype = {'ID': str})
    return df_file, df_file_GPL


#============================================================
# Cleaning Files
#============================================================

def mrna_assignment_remove_multiple_values(column):
    ''' Selecting one GeneSymbol when more than one is provided by record and
        the separator is: //
        Example: Srp54c///Srp54b///Srp54a becomes Srp54c
        input/output: pandas series
    '''
    if '//' in column:
        column = str(column)
        GeneSymbol_list = re.split(r'//', column)
        result = []
        if result == []: record = GeneSymbol_list[0]
        else: record = result[0]
    else:
        record = column
    return record


def clean_GPL_file(dataset,column_name = 'mrna_assignment' ):
    '''
```

```
    Function that clean the mrna_assignment for the GPL dataset.
    '''
    dataset[column_name] = dataset[column_name].astype(str).apply(
        mrna_assignment_remove_multiple_values)

    return dataset


#============================================================
# Merging Files
#============================================================

def populate_mrna(df_file, df_file_GPL):
    '''
    Populating the file with the GPL['mrna_assignment'] using the ID
    number as the key.
    Using the pandas .merge() method with the how='left' argument:'
    Input: two dataframes
    output: one final dataframe
    '''
    merged_df = pd.merge(df_file, df_file_GPL, how='left', left_on='ID',
                         right_on='ID')
    return merged_df

#============================================================
# Removing mRNA_Accession and renaming mrna_assignment to mRNA_Accession
#============================================================

def updating_column_names(dataset,column_remove, column_keep, new_name):
    '''
    emoving mRNA_Accession column
    and renaming mrna_assignment column to mRNA_Accession
    '''
    dataset.drop(column_remove, axis=1, inplace=True)
    dataset.rename(columns={column_keep: new_name}, inplace=True)
    return dataset


#============================================================
# Retriving Files
#============================================================

def output_file(output_file, output_path, output_file_name):
    output_file.to_csv(os.path.join(output_path, output_file_name), sep='\t')
```

0.4 Runing Main Function

- User input information manually

- Computation and outputs generated

In [3]:
```
#Main Function:

    if __name__ == "__main__":

        try:

            #============================================================
            # User input:
            #============================================================

            # path_1 = input("Enter the location of the first dataset file including:
            # file name and extension -'dp-docs.txt': \n: ")
            path_1 = '/files/GSM2386506_GEO2R_TRM_v_TN_Kupper.txt'

            #path_2 = input("Enter the location of the second dataset file (GPL file)
            # including: file name and extension -'dp-docs.txt': \n: ")
            path_2 = '/files/GPL16570-1802.txt'

            #input_skip_rows = input("Enter the number of rows to skip for the(GPL file),
            # without the headers of the dataset: \n: ")
            input_skip_rows = 19

            #input_skip_rows = input("Enter the name of the columns to keep for
            # the GPL file: \n: ")
            input_columns_to_keep = ['ID', 'mrna_assignment']

            # Removing original mRNA_Accession column and renaming mrna_assignment to
            # mRNA_Accession
            # input (dataset,column_remove, column_keep, new_name):
            column_remove = 'mRNA_Accession'
            column_keep = 'mrna_assignment'
            new_name = 'mRNA_Accession'

            #output_path = input('\nEnter the location where you want to
            # store the output file:\n ')
            output_path = '/files/input_files'
            output_file_name = 'GSM2386506_Kupper.txt'



            #============================================================
            # Computation
            #============================================================

            #computing functions
            df1 , df2 = read_files(path_1, path_2, input_skip_rows, input_columns_to_keep)
```

```
            print('Principal dataset: ')
            display(df1.head(3))
            print(df1.dtypes)

            print('----')
            print('GPL file: ')
            display(df2.head(3))
            print(df2.dtypes)
        #         print(df2.loc[df2['ID'] == 17548559])

            clean_GPL_file (df2)
            print('----')
            print('GPL file after cleaning: ')
            display(df2.head())

            print('----')
            print('Merged files: ')
            final_df = populate_mrna(df1,df2)
            display(final_df.head())

            print('----')
            print('Removing original mRNA_Accession column and renaming mrna_assignment '+
                  'to mRNA_Accession: ')
            updating_column_names(final_df,column_remove, column_keep, new_name)
            display(final_df.head())

            # output file
            output_file(final_df,output_path, output_file_name)



        except IOError:
            print ("That PATH cannot be found or does not exist.")
```

```
Principal dataset:

         ID      GeneSymbol             mRNA_Accession  adj.P.Val   P.Value  \
0  17375480        Gm14085               NM_001085518     0.0421  0.000003
1  17548559  Emp1  // Emp1   NM_010128  //  NM_010128     0.0421  0.000006
2  17266967           Ccl3                   NM_011337     0.0421  0.000008

           t         B     logFC                      SPOT_ID
0  41.250301  4.123165 -5.995670  chr2(+):122484941-122528040
1 -34.982412  3.917295  6.056938  chr6(-):135382613-135383172
```

```
2 -33.143333  3.838551  6.717706   chr11(-):83647843-83649378


ID                 object
GeneSymbol         object
mRNA_Accession     object
adj.P.Val         float64
P.Value           float64
t                 float64
B                 float64
logFC             float64
SPOT_ID            object
dtype: object
----
GPL file:


         ID                                      mrna_assignment
0  17210850  ENSMUST00000082908 // ENSEMBL // ncrna:snRNA c...
1  17210852  ENSMUST00000161581 // ENSEMBL // cdna:putative...
2  17210855  NM_008866 // RefSeq // Mus musculus lysophosph...


ID                 object
mrna_assignment    object
dtype: object
----
GPL file after cleaning:


         ID      mrna_assignment
0  17210850  ENSMUST00000082908
1  17210852  ENSMUST00000161581
2  17210855           NM_008866
3  17210869        NM_001159751
4  17210883  ENSMUST00000144339


----
Merged files:


         ID      GeneSymbol             mRNA_Accession  adj.P.Val   P.Value  \
0  17375480        Gm14085               NM_001085518      0.0421  0.000003
1  17548559  Emp1  // Emp1   NM_010128  //  NM_010128      0.0421  0.000006
2  17266967           Ccl3                   NM_011337      0.0421  0.000008
3  17385374          Nr4a2         ENSMUST00000028166      0.0421  0.000010
4  17335467         Cdkn1a                  NM_007669      0.0421  0.000010
```

```
           t         B     logFC                       SPOT_ID  \
0  41.250301  4.123165 -5.995670  chr2(+):122484941-122528040
1 -34.982412  3.917295  6.056938  chr6(-):135382613-135383172
2 -33.143333  3.838551  6.717706   chr11(-):83647843-83649378
3 -30.743830  3.718954  6.035335    chr2(-):57106830-57124003
4 -30.649314  3.713798  6.367045   chr17(+):29090979-29100722

       mrna_assignment
0        NM_001085518
1           NM_010128
2           NM_011337
3  ENSMUST00000028166
4           NM_007669
```

Removing original mRNA_Accession column and renaming mrna_assignment to mRNA_Accession:

```
         ID      GeneSymbol  adj.P.Val   P.Value          t         B  \
0  17375480        Gm14085      0.0421  0.000003  41.250301  4.123165
1  17548559  Emp1  // Emp1      0.0421  0.000006 -34.982412  3.917295
2  17266967           Ccl3      0.0421  0.000008 -33.143333  3.838551
3  17385374          Nr4a2      0.0421  0.000010 -30.743830  3.718954
4  17335467         Cdkn1a      0.0421  0.000010 -30.649314  3.713798

      logFC                       SPOT_ID       mRNA_Accession
0 -5.995670  chr2(+):122484941-122528040        NM_001085518
1  6.056938  chr6(-):135382613-135383172           NM_010128
2  6.717706   chr11(-):83647843-83649378           NM_011337
3  6.035335    chr2(-):57106830-57124003  ENSMUST00000028166
4  6.367045   chr17(+):29090979-29100722           NM_007669
```