TRM_genes_arrays

October 11, 2019

0.1 TRM_genes_arrays

- Importing files and produced a join file with ['GeneSymbol', 'mRNA_Accession', 'adj.P.Val', 'logFC'] information per file

0.1.1 Notes

- Per file: We are going to have one unique row per GeneSymbol, mean for numeric and NM_ for mRNA_Accession. We do not want EMSM (Ensamble).
- We are going to do an inner/outer join

In [1]:
```
#============================================================
# Read me
#============================================================
# TRM_genes_arrays.py
# Author: Yesika Contreras
#
# This code generated a dataframe from a list of datasets
# to be used in the modeling part
#
#
# python scripts generated 06-29-2019
# Modifications on 09-26-2019

import datetime
d = datetime. datetime. today()
print(d.strftime('%m-%d-%Y'))
```

0.2 Importing packages

In [2]:
```
#============================================================
# 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[2]: b'\x93\xf6qC\x9fH\x87!1\x08\xee\xd9\x05\x92v\xb6\xeb)\x17\xa4\x00g\x99@\x029\xd9\x90\x8
```

0.3 Defining Functions

- Defining function to be used in this script

0.3.1 Importing datasets

```
In [3]: #============================================================
        # Importing datasets
        #============================================================


        def set_directory(path_1):
            '''
            set working directory:
            '''
            os.chdir(path_1)
            path_1 = os.getcwd()
            return path_1

        def list_of_datasets(path, columns_to_keep):
            '''
            Getting a list of filenames
            (We dont need the full path as we set the directory).
```

```
        input Files are saved as '.txt'
        Converting the files into dataframes snd saving them as
        a list of dataframes.
        and adding the file name at the end of every selected column name
        '''
        # Getting a list of filenames
        all_files = [x for x in os.listdir() if x.endswith(".txt")]
        print('List of files:')
        print(all_files)
        print('------- \n ')

        # Converting the files into dataframes snd saving them as
        # a list of dataframes.
        list_of_datasets = [pd.read_csv(filename, delimiter="\t",
                                        usecols= columns_to_keep
                                       ).add_suffix('_' + filename)
                            for filename in all_files]

        #Verifying data type
        print('data type per file:')
        for file in list_of_datasets:
            print( file.dtypes)
            print('-------')
        # print(list_of_dfs[0])

        return list_of_datasets


    def dataset_size(list_of_datasets):
        for file in list_of_datasets:
            print('File name: ' + str(file.columns[0][11:]) +
                  ', ', 'Total Observations: ' +  str(file.size/4))
        print('-------')
        ## Initial Total size of observations in the list of datasets:
        print('Total size of observations in the list of datasets: given as '
              +' (observations, columns)')
        print(pd.concat(list_of_datasets, sort=False).shape )
        print('-------  \n ')
```

0.3.2 Cleaning files

```
In [4]: #============================================================
        # Cleaning files
        #============================================================

        def GeneSymbol_remove_multiple_values(column):
            ''' Selecting one GeneSymbol when more than one is provided by
            record and the separator is: ///
```

```
        Input example: Srp54c///Srp54b///Srp54a
        Rule: select a GeneSymbol that do not contain LOC, GM, #RIK
    '''
#     if re.findall("[///]", column):
    if '///' in column:
        column = str(column)
        GeneSymbol_list = re.split(r'///', column)
        patterns = ['LOC', 'GM', '#RIK']
        omit =[]
        result = []
        for i in GeneSymbol_list:
            if re.search(r"LOC", i ): omit.append(i)
            elif re.search(r"GM", i ): omit.append(i)
        result = sorted(list(set(GeneSymbol_list) - set(omit)))
        if result == []: record = GeneSymbol_list[0]
        else: record = result[0]
    else:
        record = column
    return record

# str1 = 'LOC///LOC'
# print(GeneSymbol_remove_multiple_values3(str1))


def cleaning_dataframe(df):
    '''
    input/output file is a dataframe'''
    for column in df.columns:
#         if column in ['GeneSymbol', 'mRNA_Accession']:
        if re.search('^GeneSymbol|^mRNA_Accession', column):
            #print (column)

            # Remove dupplicate string in observation per column.
            # Example: Srp54c///Srp54b///Srp54a///
            df[column] = df[column].astype(str).apply(
                GeneSymbol_remove_multiple_values)

            # Remove dupplicate string in observation per column.
            # Example: Emp1 // Emp1
            df[column] = df[column].str.split(" //", expand=True)[0]

            # remove white space
            df[column] = df[column].str.strip()

            # Replace '---' & '0' with NaN
            df[column].replace('---',np.nan, inplace=True)
            df[column].replace('0',np.nan, inplace=True)
            df[column].replace('',np.nan, inplace=True)
```

```
                df[column].replace('nan',np.nan, inplace=True)

        return df
```

0.3.3 Identifying Missing values

```
In [5]: #============================================================
        # Identifying Missing values
        #============================================================

        def percentage_missing_values(list_of_datasets):
            '''
            Identifying percentage of missing values given a list of dataframes
            '''
            for file in list_of_datasets:
                print ( round(file.isna().sum() *100 / (file.size/4) ,2) )
                print('---------')


        #Removing missing values for the GeneSymbol columns.
        def drop_missing_values(list_of_datasets):
            '''
            If there are missing values on the GeneSymbol, then we drop the row.
            Otherwise we keep the row.
            '''
            for file in list_of_datasets:
                file.dropna(subset = [file.filter(regex='^GeneSymbol',
                                                  axis=1).columns[0]], inplace=True)
        #     file.dropna(inplace=True)
            return list_of_datasets


        # Imputing missing values
        def impute_missing_values(list_of_datasets):
            '''
            If there are missing values on the numeric fields, then impute with the mean.
            If there are missing values on the categorical fields, then impute with the mode.
            '''
            for file in list_of_datasets:

                GeneSymbol_col = file.filter(regex='^GeneSymbol',axis=1).columns[0]
                mRNA_col = file.filter(regex='^mRNA_Accession',axis=1).columns[0]
                adj_P_Val_col = file.filter(regex='^adj.P.Val',axis=1).columns[0]
                logFC_col = file.filter(regex='^logFC',axis=1).columns[0]

                file[mRNA_col] = file.groupby(GeneSymbol_col)[mRNA_col].transform(
                    lambda x: x.fillna(x.mode().get(0,'NaN/#N/A')))
                file[adj_P_Val_col] = file.groupby(GeneSymbol_col)[adj_P_Val_col].transform(
```

```
                lambda x: x.fillna(x.mean()))
            file[logFC_col] = file.groupby(GeneSymbol_col)[logFC_col].transform(
                lambda x: x.fillna(x.mean()))

        return list_of_datasets
```

0.3.4 Removing Duplicate Records

```
In [6]: #============================================================
        # Removing duplicate records (Duplicate rows having the same GeneSymbol)
        #============================================================

        ## Removing duplicate records (rows)
        def duplicate_rows_one_column (data):
            '''
            Remove duplicate rows having the same GeneSymbol
            Calculate mean for numeric columns
            Maintain mRNA_Accession NM
            input/output file is a dataframe
            '''
            df_duplicate_rows = data.groupby([data.filter(regex='^GeneSymbol',
                                                          axis=1).columns[0]],
                                             as_index=False).aggregate(
                {data.filter(regex='^mRNA_Accession',axis=1).columns[0]: 'max',
                 data.filter(regex='^adj.P.Val',axis=1).columns[0]: 'mean',
                 data.filter(regex='^logFC',axis=1).columns[0]: 'mean'})

            return df_duplicate_rows
        # data.filter(regex='^mRNA_Accession',axis=1).columns[0]: pd.Series.mode,
```

0.3.5 Merging Multiple dataframes

```
In [7]: #============================================================
        # Joinining Dataframes
        #============================================================

        ## Merging the files using merge and reduce function and after
        # compiling the list of dataframes to merge.
        # To keep the values that belong to the same gene symbol we need
        # to merge it on the GeneSymbol.
        # We are doing an outer join (NAs will be added)


        def merging_list_of_datasets(list_of_datasets, join_type = 'outer'):

            df_merged = reduce(lambda  left, right:
                               pd.merge(left, right,
                                        left_on = left.filter(
```

```
                                            regex='^GeneSymbol',axis=1).columns[0] ,
                                        right_on= right.filter(
                                            regex='^GeneSymbol',axis=1).columns[0],
                                        how = join_type),
                                list_of_datasets)

        return df_merged


    #If column multi indexes
    #(it was injecting the 'on' as a column which worked for the first merge,
    # but subsequent merges failed),
    #instead I got it to work with:
    #df = reduce(lambda left, right: left.join(right, how='outer', on='Date'), dfs)
```

0.3.6 Retriving Files

In [8]:
```
#============================================================
# Retriving Files
#============================================================

def output_file(output_file, output_path, output_file_name):
    '''
    Retrive a single file,
    inputs output_file: dataframe,
    output_path:folder location,
    output_file_name: file name with .txt extension
    '''
    output_file.to_csv(os.path.join(output_path, output_file_name), sep='\t')

def output_list_of_datasets(output_list_of_datasets, output_path,
                            output_file_name):
    '''
    Retrive a list_of_datasets,
    inputs output_file: list_of_datasets,
    output_path:folder location,
    output_file_name: file name with .txt extension
    '''
    pd.concat(output_list_of_datasets, sort=True).to_csv(
        os.path.join(output_path, output_file_name), sep='\t')
```

In [ ]:
```
```

0.4 Runing Main Functions

- User input information manually
- Computation and outputs generated

In [9]:
```
#============================================================
# User input:
#============================================================

#path = input("Enter the folder location of the dataset files:")
path = '/TRM_GeneArrays/input_files'

#columns_to_keep = input("Enter the name of the columns to keep for GPL file:")
columns_keep = ['GeneSymbol', 'mRNA_Accession', 'adj.P.Val', 'logFC']

#output_path = input('Enter the location to store the output file:')
output_path = '/TRM_GeneArrays/output_files'
```

In [10]:
```
#============================================================
# Computation Importing datasets
#============================================================

# setting working directory:
path = set_directory(path)

# Converting files into dataframes
list_of_dfs = list_of_datasets(path, columns_keep)

# Original datasets sizes
print('\nOriginal dataset size: ')
dataset_size(list_of_dfs)
```

```
List of files:
['GSM2386506_Kupper.txt', 'GSE47045_GEO2R_TRM_v_TN_Carbone.txt', 'GSE79858_Slansky_TIL_v_TN.tx
-------

data type per file:
GeneSymbol_GSM2386506_Kupper.txt         object
adj.P.Val_GSM2386506_Kupper.txt         float64
logFC_GSM2386506_Kupper.txt             float64
mRNA_Accession_GSM2386506_Kupper.txt     object
dtype: object
-------
GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt         object
mRNA_Accession_GSE47045_GEO2R_TRM_v_TN_Carbone.txt     object
adj.P.Val_GSE47045_GEO2R_TRM_v_TN_Carbone.txt         float64
logFC_GSE47045_GEO2R_TRM_v_TN_Carbone.txt             float64
dtype: object
-------
GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt         object
mRNA_Accession_GSE79858_Slansky_TIL_v_TN.txt     object
adj.P.Val_GSE79858_Slansky_TIL_v_TN.txt         float64
logFC_GSE79858_Slansky_TIL_v_TN.txt             float64
```

```
dtype: object
-------
GeneSymbol_GSE_Anderson_TIL_v_Naive.txt         object
mRNA_Accession_GSE_Anderson_TIL_v_Naive.txt     object
adj.P.Val_GSE_Anderson_TIL_v_Naive.txt         float64
logFC_GSE_Anderson_TIL_v_Naive.txt             float64
dtype: object
-------

Original dataset size:
File name: GSM2386506_Kupper.txt,  Total Observations: 41345.0
File name: GSE47045_GEO2R_TRM_v_TN_Carbone.txt,  Total Observations: 34761.0
File name: GSE79858_Slansky_TIL_v_TN.txt,  Total Observations: 35557.0
File name: GSE_Anderson_TIL_v_Naive.txt,  Total Observations: 45101.0
-------
Total size of observations in the list of datasets: given as (observations, columns)
(156764, 16)
-------
```

```
In [11]: #============================================================
         # Computation Cleaning datasets
         #============================================================
         print('Checking values before cleaning: ')
         print(list_of_dfs[1]['GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt'][34732])

         for file in list_of_dfs:
             cleaning_dataframe(file)

         print('\nChecking values after cleaning: ')
         print(list_of_dfs[1]['GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt'][34732])
         print('---------')
```

```
Checking values before cleaning:
Vmn2r51///Vmn2r36///Vmn2r34///Vmn2r28///Vmn2r48///Vmn2r35///Vmn2r33///Vmn2r43///Vmn2r29///Vmn2

Checking values after cleaning:
Vmn2r28
---------
```

```
In [12]: #============================================================
         # Computation Identifying Missing values
         #============================================================

         ## Percentage of missing values
         print('\nPercentage of missing values: ')
```

```
percentage_missing_values(list_of_dfs)

#  Let's figure out which values are missing!
print('\nExample of missing values for GeneSymbol_GSM2386506_Kupper: ')
print( list_of_dfs[0][ list_of_dfs[0]['GeneSymbol_GSM2386506_Kupper.txt'].isnull()
                     ].head(3) )
print('---------')

# Removing missing values for the GeneSymbol columns.
drop_missing_values(list_of_dfs)
list_of_dfs = impute_missing_values(list_of_dfs)

## Checking the new Percentage of missing values
print('\nNew percentage of missing values: ')
percentage_missing_values(list_of_dfs)


# Datasets size after removing missing values:
print('\nDatasets size after removing missing values: ')
dataset_size(list_of_dfs)

# Saving the list of datasets  after the cleaning step
file_name_cleaned = 'cleaned_files.txt'
output_list_of_datasets(list_of_dfs, output_path, file_name_cleaned)
```

```
Percentage of missing values:
GeneSymbol_GSM2386506_Kupper.txt        35.45
adj.P.Val_GSM2386506_Kupper.txt          0.00
logFC_GSM2386506_Kupper.txt              0.00
mRNA_Accession_GSM2386506_Kupper.txt    18.61
dtype: float64
---------
GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt        29.67
mRNA_Accession_GSE47045_GEO2R_TRM_v_TN_Carbone.txt    23.16
adj.P.Val_GSE47045_GEO2R_TRM_v_TN_Carbone.txt          0.00
logFC_GSE47045_GEO2R_TRM_v_TN_Carbone.txt              0.00
dtype: float64
---------
GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt        29.63
mRNA_Accession_GSE79858_Slansky_TIL_v_TN.txt    18.45
adj.P.Val_GSE79858_Slansky_TIL_v_TN.txt          0.00
logFC_GSE79858_Slansky_TIL_v_TN.txt              0.00
dtype: float64
---------
GeneSymbol_GSE_Anderson_TIL_v_Naive.txt        11.62
mRNA_Accession_GSE_Anderson_TIL_v_Naive.txt     0.14
adj.P.Val_GSE_Anderson_TIL_v_Naive.txt          0.00
```

```
logFC_GSE_Anderson_TIL_v_Naive.txt              0.00
dtype: float64
---------

Example of missing values for GeneSymbol_GSM2386506_Kupper:
   GeneSymbol_GSM2386506_Kupper.txt  adj.P.Val_GSM2386506_Kupper.txt  \
16                              NaN                           0.0421
35                              NaN                           0.0421
36                              NaN                           0.0421

    logFC_GSM2386506_Kupper.txt mRNA_Accession_GSM2386506_Kupper.txt
16                    -3.759582                   ENSMUST00000102339
35                     3.472884                                  NaN
36                    -3.747909                                  NaN
---------

New percentage of missing values:
GeneSymbol_GSM2386506_Kupper.txt        0.0
adj.P.Val_GSM2386506_Kupper.txt         0.0
logFC_GSM2386506_Kupper.txt             0.0
mRNA_Accession_GSM2386506_Kupper.txt    0.0
dtype: float64
---------
GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt        0.0
mRNA_Accession_GSE47045_GEO2R_TRM_v_TN_Carbone.txt    0.0
adj.P.Val_GSE47045_GEO2R_TRM_v_TN_Carbone.txt         0.0
logFC_GSE47045_GEO2R_TRM_v_TN_Carbone.txt             0.0
dtype: float64
---------
GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt        0.0
mRNA_Accession_GSE79858_Slansky_TIL_v_TN.txt    0.0
adj.P.Val_GSE79858_Slansky_TIL_v_TN.txt         0.0
logFC_GSE79858_Slansky_TIL_v_TN.txt             0.0
dtype: float64
---------
GeneSymbol_GSE_Anderson_TIL_v_Naive.txt        0.0
mRNA_Accession_GSE_Anderson_TIL_v_Naive.txt    0.0
adj.P.Val_GSE_Anderson_TIL_v_Naive.txt         0.0
logFC_GSE_Anderson_TIL_v_Naive.txt             0.0
dtype: float64
---------

Datasets size after removing missing values:
File name: GSM2386506_Kupper.txt,  Total Observations: 26688.0
File name: GSE47045_GEO2R_TRM_v_TN_Carbone.txt,  Total Observations: 24447.0
File name: GSE79858_Slansky_TIL_v_TN.txt,  Total Observations: 25023.0
File name: GSE_Anderson_TIL_v_Naive.txt,  Total Observations: 39860.0
-------
```

```
Total size of observations in the list of datasets: given as (observations, columns)
(116018, 16)
-------
```

In [13]:
```
# Checking unique values per GeneSymbol before running the function.

for file in list_of_dfs:
#     print (file[file.filter(regex='^GeneSymbol',axis=1).columns[0]
# ].astype(object).nunique() )
    print( file[file.filter(regex='^GeneSymbol',axis=1).columns[0]
              ].astype(object).describe() )
```

```
count     26688
unique    24389
top       Syne1
freq         59
Name: GeneSymbol_GSM2386506_Kupper.txt, dtype: object
count     24447
unique    21637
top       Hmcn1
freq         77
Name: GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt, dtype: object
count        25023
unique       21639
top       Snord115
freq            79
Name: GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt, dtype: object
count         39860
unique        21415
top       Igh-VJ558
freq             17
Name: GeneSymbol_GSE_Anderson_TIL_v_Naive.txt, dtype: object
```

In [14]:
```
#============================================================
# Computation Removing duplicate records
#============================================================

## creating a new list of dataframes after removing duplicate records
new_list_of_dfs = []
for file in list_of_dfs:
    new_list_of_dfs.append(duplicate_rows_one_column(file))


# Datasets size after removing duplicates in GeneSymbol
print('\nDatasets size after removing duplicates in GeneSymbol: ')
```

```
        dataset_size(new_list_of_dfs)

        # Saving the list of datasets after removing duplicates
        file_name_duplicates= 'removed_duplicates_files.txt'
        output_list_of_datasets(new_list_of_dfs, output_path, file_name_duplicates)
```

```
Datasets size after removing duplicates in GeneSymbol:
File name: GSM2386506_Kupper.txt,  Total Observations: 24389.0
File name: GSE47045_GEO2R_TRM_v_TN_Carbone.txt,  Total Observations: 21637.0
File name: GSE79858_Slansky_TIL_v_TN.txt,  Total Observations: 21639.0
File name: GSE_Anderson_TIL_v_Naive.txt,  Total Observations: 21415.0
-------
Total size of observations in the list of datasets: given as (observations, columns)
(89080, 16)
-------
```

```
In [15]: #============================================================
         # Computation Merging multiple dataframes
         #============================================================

         df_merged_outer = merging_list_of_datasets(new_list_of_dfs, join_type = 'outer')
         print('\nMerged dataframe with outer join')
         print('Total size of observations in the outer dataframe given as'
               +' (observations, columns): ')
         print(df_merged_outer.shape)
         display(df_merged_outer.head(3))
         print('---------')

         df_merged_inner = merging_list_of_datasets(new_list_of_dfs, join_type = 'inner')
         print('\nMerged dataframe with inner join')
         print('Total size of observations in the inner dataframe given as'
               +' (observations, columns): ')
         print(df_merged_inner.shape)
         display(df_merged_inner.head(3))
```

```
Merged dataframe with outer join
Total size of observations in the outer dataframe given as (observations, columns):
(33610, 16)
```

	GeneSymbol_GSM2386506_Kupper.txt	mRNA_Accession_GSM2386506_Kupper.txt	\
0	0610005C13Rik	NR_038166	
1	0610007C21Rik	NM_027855	
2	0610007L01Rik	NM_027854	

```
   adj.P.Val_GSM2386506_Kupper.txt  logFC_GSM2386506_Kupper.txt  \
0                           0.8534                    -0.120814
1                           0.6331                     0.290360
2                           0.9048                     0.119549

  GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                            NaN
1                                            NaN
2                                            NaN

  mRNA_Accession_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                                NaN
1                                                NaN
2                                                NaN

   adj.P.Val_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                            NaN
1                                            NaN
2                                            NaN

   logFC_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                        NaN
1                                        NaN
2                                        NaN

  GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt  \
0                                    NaN
1                                    NaN
2                                    NaN

  mRNA_Accession_GSE79858_Slansky_TIL_v_TN.txt  \
0                                       NaN
1                                       NaN
2                                       NaN

   adj.P.Val_GSE79858_Slansky_TIL_v_TN.txt  \
0                                    NaN
1                                    NaN
2                                    NaN

   logFC_GSE79858_Slansky_TIL_v_TN.txt  \
0                                 NaN
1                                 NaN
2                                 NaN

  GeneSymbol_GSE_Anderson_TIL_v_Naive.txt  \
0                           0610005C13Rik
1                                     NaN
```

```
2                                    NaN

  mRNA_Accession_GSE_Anderson_TIL_v_Naive.txt  \
0                                   NR_038165
1                                         NaN
2                                         NaN

   adj.P.Val_GSE_Anderson_TIL_v_Naive.txt  logFC_GSE_Anderson_TIL_v_Naive.txt
0                                   0.792                             -0.06301
1                                     NaN                                  NaN
2                                     NaN                                  NaN


---------

Merged dataframe with inner join
Total size of observations in the inner dataframe given as (observations, columns):
(16101, 16)


  GeneSymbol_GSM2386506_Kupper.txt mRNA_Accession_GSM2386506_Kupper.txt  \
0                     0610007P14Rik                              NM_021446
1                     0610009B22Rik                               BC024353
2                     0610009L18Rik                              NR_038126

   adj.P.Val_GSM2386506_Kupper.txt  logFC_GSM2386506_Kupper.txt  \
0                           0.9545                    -0.089542
1                           0.9759                     0.082806
2                           0.8914                     0.155382

  GeneSymbol_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                   0610007P14Rik
1                                   0610009B22Rik
2                                   0610009L18Rik

  mRNA_Accession_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                          NM_021446
1                                          NM_025319
2                                          NR_038126

   adj.P.Val_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                       0.430283
1                                       0.680325
2                                       0.983132

   logFC_GSE47045_GEO2R_TRM_v_TN_Carbone.txt  \
0                                  -1.062394
1                                   0.423292
```

```
2                                -0.023245

  GeneSymbol_GSE79858_Slansky_TIL_v_TN.txt  \
0                            0610007P14Rik
1                            0610009B22Rik
2                            0610009L18Rik

  mRNA_Accession_GSE79858_Slansky_TIL_v_TN.txt  \
0                                    NM_021446
1                                    NM_025319
2                                    NR_038126

   adj.P.Val_GSE79858_Slansky_TIL_v_TN.txt  \
0                                 0.029688
1                                 0.002676
2                                 0.097500

   logFC_GSE79858_Slansky_TIL_v_TN.txt  \
0                               0.973
1                               1.430
2                              -0.363

  GeneSymbol_GSE_Anderson_TIL_v_Naive.txt  \
0                            0610007P14Rik
1                            0610009B22Rik
2                            0610009L18Rik

  mRNA_Accession_GSE_Anderson_TIL_v_Naive.txt  \
0                                    NM_021446
1                                    NM_025319
2                                    NR_038126

   adj.P.Val_GSE_Anderson_TIL_v_Naive.txt  logFC_GSE_Anderson_TIL_v_Naive.txt
0                                  0.0292                            0.833661
1                                  0.7920                            0.217723
2                                  0.2900                            0.254251
```

In [16]:
```
# Saving the merged dataframes
outer_file_name = 'df_merged_outer.txt'
inner_file_name = 'df_merged_inner.txt'
output_file(df_merged_outer, output_path, outer_file_name)
output_file(df_merged_inner, output_path, inner_file_name)
```