Bioinfo_Project User Manual

The increase availability of genomic data poses the question regarding the extraction of relevant biological features from genomic datasets. For this reason, this project aims to find features selection methods that permits to infer relevant biological information from mRNA and miRNA expression data of kidney cancer. The three main parts of this assignment are:

1. *Download and pre-processing*
2. *Features selection methods*
3. *Classification and validation*

The guide manual is then presented to correctly perform all the steps.

1) Download and pre-processing

The first part of the project is used to download the files of those affected by kidney cancer who have both mRNA and miRNA data. Then the matrices are created with the genes (features) on the columns and the subject's tumor type (labels) on the lines. Finally, the insignificant features are removed and the ‘zero-mean’ and ‘min-max’ rescale is done.

1.1) creazione_manifest.py

This script is responsible for processing the json and manifest files related to mRNA and miRNA, in order to produce the final manifest files that are used to download the data from the server.

The following global variables are used: pay attention to the names of the files used in the script and present in the folder.

```
mrnaJson = '2_mRNA.json'
mrnaManifest = '2_mRNA_manifest.txt'
mirnaJson = '1_miRNA.json'
mirnaManifest = '1_miRNA_manifest.txt'
```

Output:

```
nuovo_manifest_m.txt
nuovo_manifest_mi.txt
new_label_mi.txt
new_label_m.txt
```

1.2) Data download using the client gdc_client.exe

With the following commands (run as administrator) the files contained in **new_manifest_m.txt** and **new_manifest_mi.txt** are downloaded.

```
gdc-client.exe download -m
<absolute_path_to_project_folder>\n1_unione_maninfest\nuovo_manifest_m.txt --dir
.\mRNA\
```

```
gdc-client.exe download -m
<absolute_path_to_project_folder>\n1_unione_maninfest\nuovo_manifest_mi.txt --dir
.\miRNA\
```

N.B.: Check the confirmation message of "Successfully downloaded: 1160".

1.3) Dataset creation

- Move the files ”new_label_mi.txt” and ”new_label_m.txt” (which contain the labels) created in step(1) to the folder ./n2_creazione_dataset.
- Run scripts ”move_files_miRNA.py” and ”move_files_mRNA.py”.
- Run script ”creazione_dataset.py”.

Output:

```
dataset_miRNA.csv
dataset_mRNA.csv
```

1.4) Dataset preprocessing

Run the script dataset_Preprocessing.py twice, changing from time to time the parameter (the dataset filename) given to the script.
Input commands:

- dataset_Preprocessing.py dataset_miRNA.csv
- dataset_Preprocessing.py dataset_mRNA.csv

Output:

```
scaled_dataset_miRNA.csv
scaled_dataset_mRNA.csv
```

2) Feature Selection Methods

Two different approaches have been chosen for the data features selection.

1. *Classic:* where the FS is carried out on each dataset
2. *multi-view:* where the FS is carried out considering the two datasets together.

For the first approach, the methods considered most suitable in the literature for this type of data were chosen:

- Random Forest mean decrease impurity
- Deision tree
- Xgboost
- KNeighborsRegressor
- RFECV

Given the high computational cost, the latter two methods are not recommended for mRNA data.

For the second approach, a method based on the canonical correlation analysis (CCA) was implemented.

2.1) copyDatasetfromPath.py

Every time you need to move multiple dataset files into the *n4_featureSelectionMet*hods folder in order to use them for feature selection methods, run *copyDatasetfromPath.py* to move every ".csv" file from the source path given as input to the function.

2.2) Running FS methods

Run the script related to the desired FS method and answer the questions asked on the command line: the answers given by the user perform the method with different parameters.

```
a) Complete or reduced list of labels? (R = reduced/C = complete)
b) Which dataset do you want to process?
c) Do you want to save the dataset containing only the most relevant features?
(Y/N)
d) How many features do you want to select?
```

a) Reduced = implements the FS on the dataset containing a smaller number of labels without considering all subcategories of tumors.
Complete = implements the FS on the dataset containing all types of labels.
b) Name of the dataset to use.
c) Y = saves a dataset that contains only the most relevant features obtained from the FS method used.

N = executes the FS method without saving the dataset.
d) Number of features to select (1-1000).

Output:

```
<FS>_<nFeatures>_<labelType>_dataset.csv
Result_FS.txt
```

2.3) Canonical Correlation Analysis

To implement a multi-view features selection, a CCA based method is implemented. After selecting the two datasets and the number of features to extract it can be possible to choose between the unsupervised and supervised version of the function. The output will be a new dataset with the top features selected from each input dataset.

```
a) Dataset#1
b) Dataset#2
c) Number of features
d) Supervised version?
e) Save CCA dataset?
```

Output:

```
CCA_result_dataset.csv
```

3) Classification and validation

Finally, to demonstrate the quality of the features found, a classifier and validation based on literature data was performed.

3.1) Convolutional Neural Network

In order to evaluate the goodness of the feature selection method a CNN is used for the classification of the data. In particular in "CNN" script is present:

- 'Myplot' class for the plot of the confusion matrix.
- 'Define the model' in which the CNN is defined.
- 'Import Data' where the dataset is divided in train and test set
- 'Classification' for the train of the CNN, classification of test set and plot the confusion matrix

3.2) Validation

A method to evaluate the goodness of the features selection is to check in literature if the mRNA and miRNA selected are involved in the renal cancer. As regards the mRNA genes, the "Atlas of human proteins" (https://www.proteinatlas.org/) was used to verify if they were prognostic. For miRNAs, on the other hand, not having found a dataset that provided the desired information, a literature search was done to find studies that demonstrated the involvement of some miRNAs in kidney cancer[1-7].

In order to validate features, three scripts are used:

a) fromCSVtoFeatures.py: gets as input the dataset filename (.csv) and returns two output files, *<orignal_dataset_name>_20_mRNA.txt* and *<orignal_dataset_name>_20_miRNA.txt*, each of which contains the list of the top 20 features of mRNA/miRNA.

b) getMirnaList.py: gets as input two files:
   - text file with the best miRNA features found through a literature search.
   - *<orignal_dataset_name>_20_miRNA.txt created at point (a).*

c) common gene names.py: gets as input the *<orignal_dataset_name>_20_mRNA.txt* file. As output returns a file with the gene names translated, and opens a tab in the browser for each gene to verify in proteinatlas.org website.

Reference

- Ka-Lok Ng, View ORCID ProfileY-h Taguchi Identification of miRNA signatures for kidney renal clear cell carcinoma using the tensor-decomposition method
- hiomi Ei, Sugai Tamotsu Analysis of Expression Patterns of MicroRNAs That Are Closely Associated With Renal Carcinogenesis
- Trilla-Fuertes L, Miranda N, Castellano D, López-Vacas R, Farfán Tello CA, et al. (2020) miRNA profiling in renal carcinoma suggest the existence of a group of pro-angionenic tumors in localized clear cell renal carcinoma.
- Guanghui Ying Ruilan Wu Identification of eight key miRNAs associated with renal cell carcinoma: A meta-analysis
- Matthew D Young, Thomas J Mitchell, Cellular mRNA signals in human kidney tumors

- Liangyou Gu1,*, Hongzhao Li1 MicroRNAs as prognostic molecular signatures in renal cell carcinoma: a systematic review and meta-analysis
- Mariagrazia Granata1, Lorenzo Malatino I microRNA nelle principali patologie renali: una nuova frontiera per il nefrologo