C. Classification

C.1. Unsupervised classification

How can we detect groups of patients with similar expression profile? What microRNAs or genes have a common intensity pattern for an experimental group? Could we explore our data before continuing the analysis?

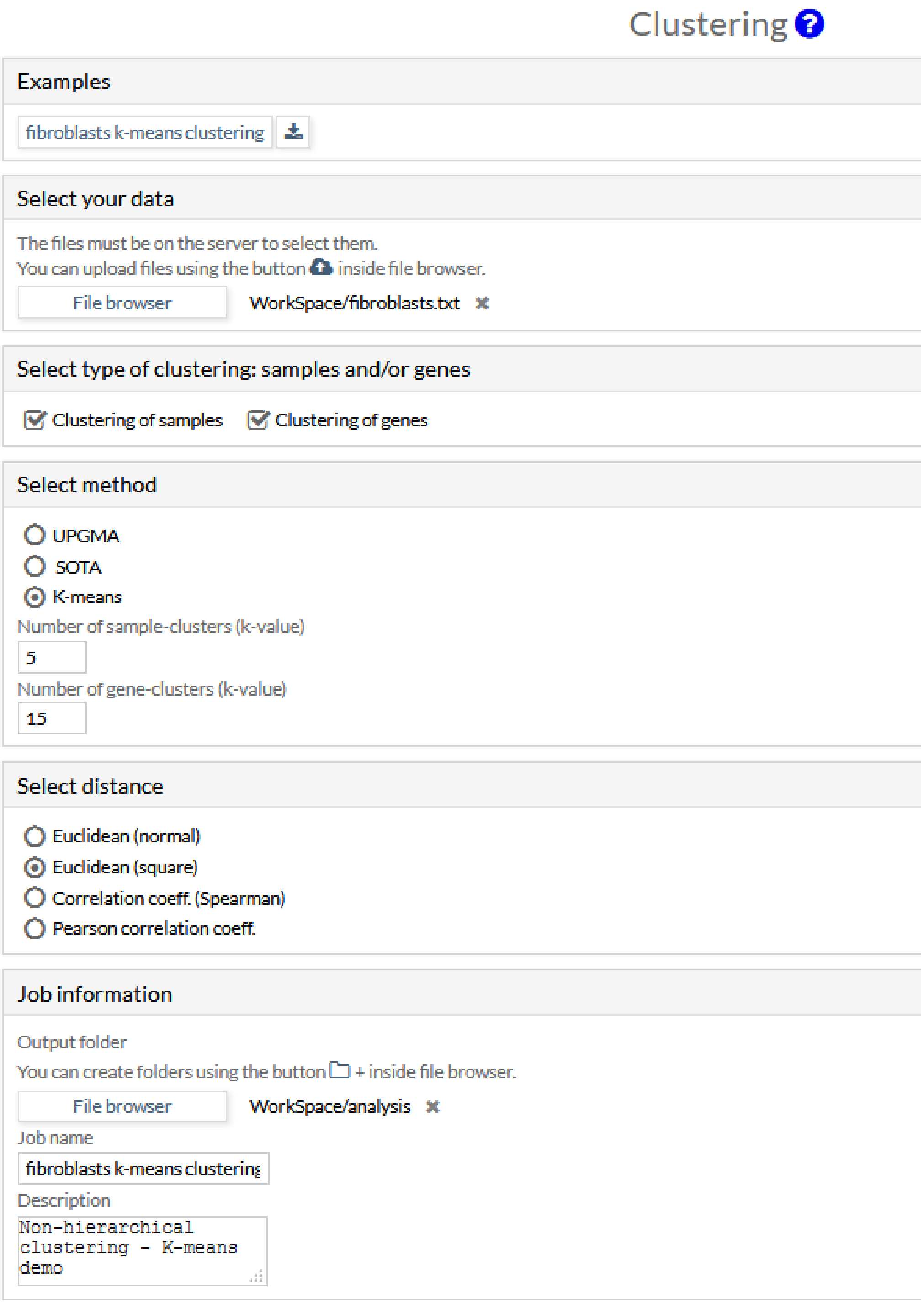
Activity 1. Online example

1. Go to the Babelomics page and select the *Clustering* option from the *Expression menu*.
2. Press the online example and you will see how the parameters and form fields are now filled. As you can notice, this example is prepared to perform a clustering analysis on genes (rows) and conditions (columns) using the K-means algorithm with 5 sample-clusters and 15 gene-clusters. Here, the selected distance is Euclidean (square).
3. Press Launch job, and wait for your job to be finished.
4. When the process finishes, a new blue job is shown at the right side of the web page. Press it to check your results.

Questions

These are some questions that you should be able to answer about the previous example:

1. Do you think that the clustering was able to differentiate any group of coexpressed genes?
2. How many sample clusters are there? and gene clusters?
3. Launch this online example using different clustering methods and compare the results. Which are the differences between the results of these results for different methods?
4. What about newick format?



Clustering

Examples

fibroblasts k-means clustering

Select your data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/fibroblasts.txt ✖

Select type of clustering: samples and/or genes

☑ Clustering of samples ☑ Clustering of genes

Select method

○ UPGMA
○ SOTA
● K-means

Number of sample-clusters (k-value)

5

Number of gene-clusters (k-value)

15

Select distance

○ Euclidean (normal)
● Euclidean (square)
○ Correlation coeff. (Spearman)
○ Pearson correlation coeff.

Job information

Output folder

You can create folders using the button + inside file browser.

File browser WorkSpace/analysis ✖

Job name

fibroblasts k-means clusterinç

Description

```
Non-hierarchical
clustering - K-means
demo
```

Job information

Name : *clustering_act1*

Description : *Non-hierarchical clustering - K-means demo*

Tool : *clustering*

Output folder : *WorkSpace/analysis/20190308152446/*

Input parameters

Dataset file name: *fibroblasts.txt*

Clustering of: *samples, genes*

Method: *kmeans, k-value (samples clustering) = 5, k-value (genes clustering) = 15*

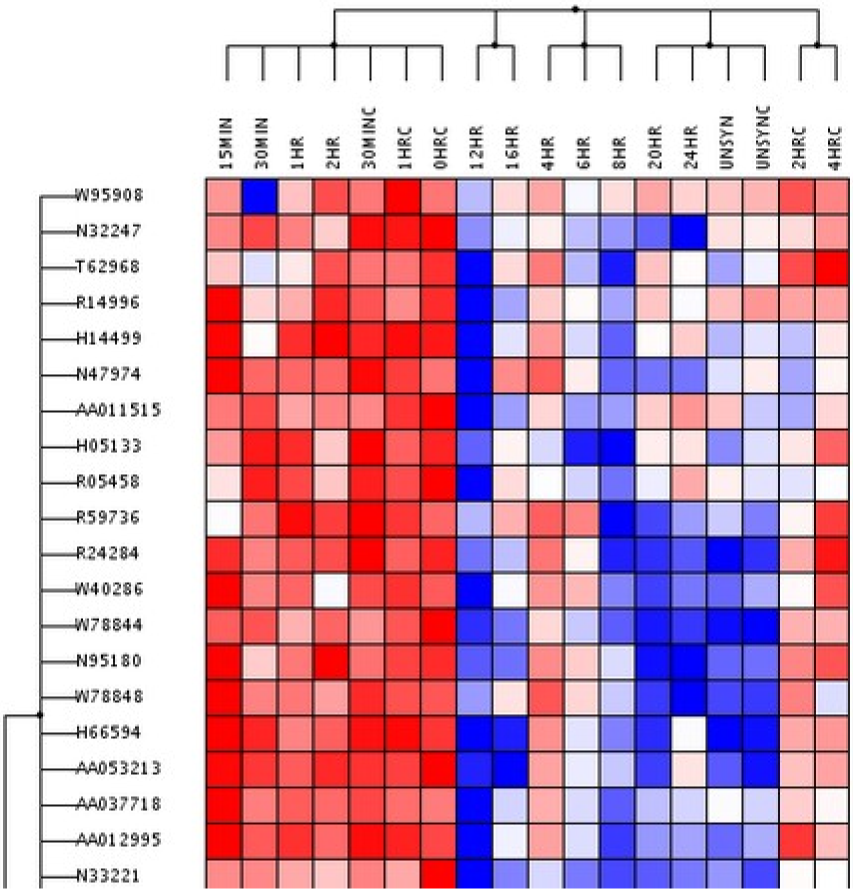
Distance: *square*

Clusters in newick format

Clusters of genes *genes.nw*

Clusters of samples *samples.nw*

Cluster images



Job information

Name : *clustering_act1_sota_eu2*

Description : *Non-hierarchical clustering - K-means demo*

Tool : *clustering*

Output folder : *WorkSpace/analysis/20190308152450/*

Input parameters

Dataset file name: *fibroblasts.txt*

Clustering of: *samples, genes*

Method: *sota*

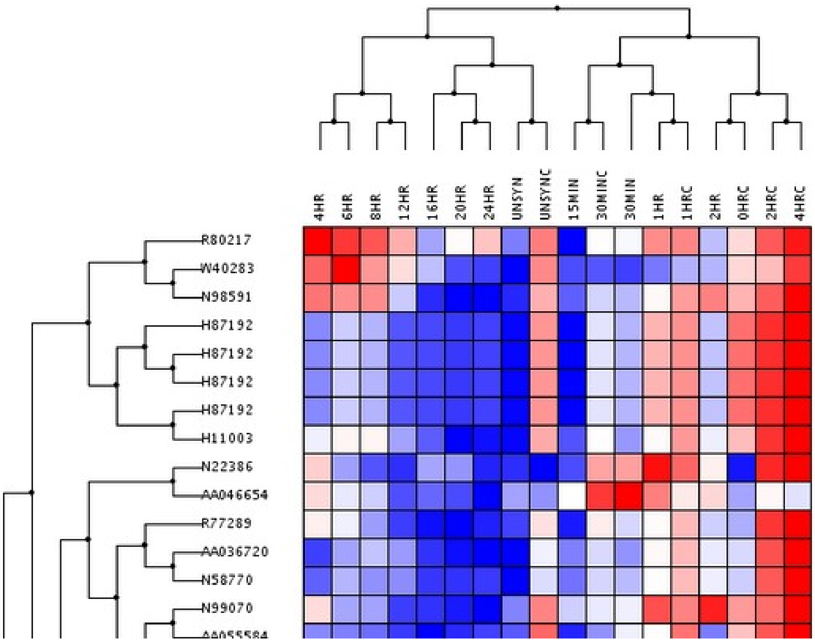
Distance: *square*

Clusters in newick format

Clusters of genes [*genes.nw*]

Clusters of samples [*samples.nw*]

Cluster images



ACTIVITY 2.

Clustering analysis for expression data in arthritis

The etiology of **rheumatoid arthritis** is not known with certainty. In order to generate information that clarifies this point, a study of expression microarrays has been proposed, which will allow characterizing this disease at the molecular level and finding some key mechanisms that will improve its prevention and treatment.

Goal

Detect homogeneous groups of subjects according to their transcriptomic profile and evaluate the possible presence of anomalous patterns.

Data

We have normalized data from Affymetrix microarrays for three experimental groups:

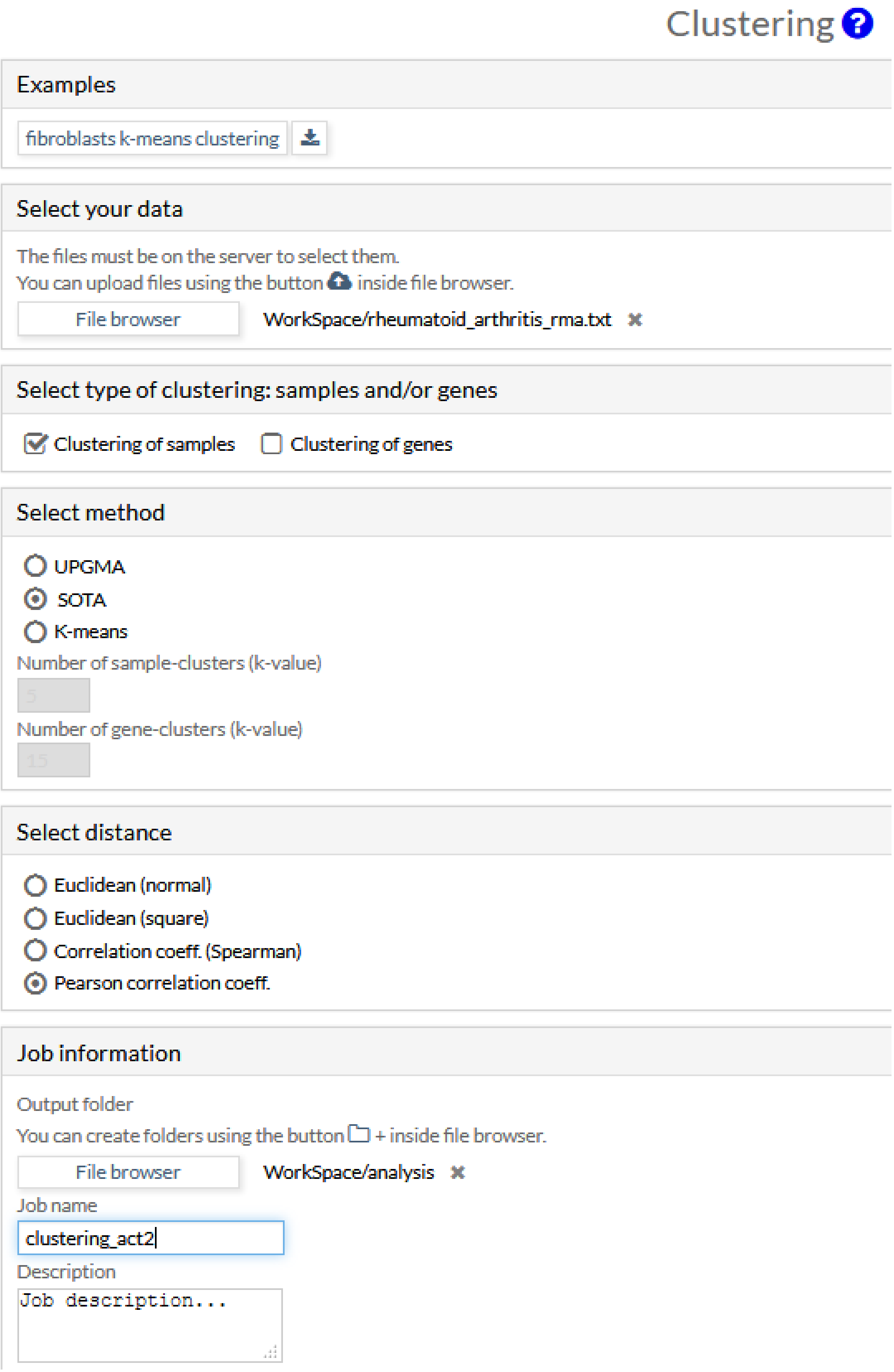
- 5 patients with rheumatoid arthritis (RA1-RA5).
- 4 patients with osteoarthritis (OA1-OA4).
- 6 healthy people (H1-H6).

Work plan

1. Open the data file of **gene expression** with a spreadsheet and inspect its contents. There will be as many columns as subjects and as many rows as genes.
2. Upload this txt file in Babelomics from the “Upload” menu. We will have to indicate the type of data that we upload: “Data matrix expression”. This link describes the different types of data that we can use in Babelomics: https://github.com/babelomics/babelomics/wiki/Data-types.
3. Next, we select the clustering by samples. We chose the “SOTA” clustering method and the distance “Pearson correlation coefficient”. We assign a name to the job and execute it.
4. Perform a clustering for genes (to begin with, those that are by default). We assign a name to the job and execute it.

Questions

1. Are there groups of samples with a similar transcriptomic profile? How many groups appear?
2. Is there any sample that has an anomalous behavior when comparing with other subjects? Any proposal?
3. Do you think that if we performed a differential expression analysis we would obtain a large number of differentially expressed genes?
4. Any incidence with clustering by genes?



Clustering ?

Examples

fibroblasts k-means clustering

Select your data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/rheumatoid_arthritis_rma.txt ✖

Select type of clustering: samples and/or genes

☑ Clustering of samples □ Clustering of genes

Select method

○ UPGMA
● SOTA
○ K-means

Number of sample-clusters (k-value)

5

Number of gene-clusters (k-value)

15

Select distance

○ Euclidean (normal)
○ Euclidean (square)
○ Correlation coeff. (Spearman)
● Pearson correlation coeff.

Job information

Output folder

You can create folders using the button + inside file browser.

File browser WorkSpace/analysis ✖

Job name

clustering_act2

Description

Job description...

Job information

Name : *clustering_act2_sota_pearson*

Description : *Job description...*

Tool : *clustering*

Output folder : *WorkSpace/analysis/20190308152747/*

Input parameters

Dataset file name: *rheumatoid_arthritis_rma.txt*

Clustering of: *samples*

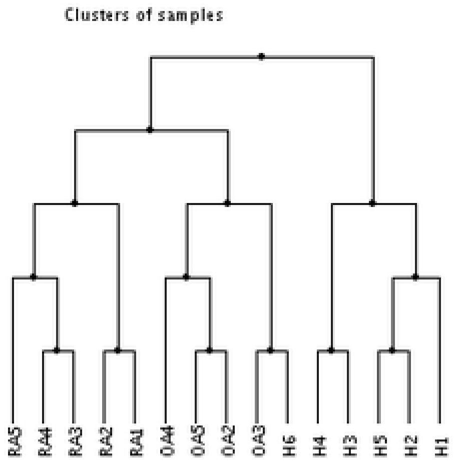
Method: *sota*

Distance: *pearson*

Clusters in newick format

Clusters of samples *samples.nw*

Cluster images



Warnings

Warning: *This release limits the heatmap in tree images to 1000 genes*

ACTIVITY 3. RNA-Seq data analysis: unsupervised classification or clustering

Goal

Detect homogenous groups of subjects according to their transcriptomic profile.

Data

We are studying a complex disease in which we know that a certain hormone has an important role. For them, we designed an experiment with RNA-Seq in mice with two groups: 6 wild type mice (WT) and 6 mice treated with T3 hormone.

These data were obtained after applying a primary analysis that included the evaluation of the quality of the sequences, mapping and quantification of expression at the gene level. We have expression levels (non-normalized counts) for the 12 mice described in 38,293 genes.

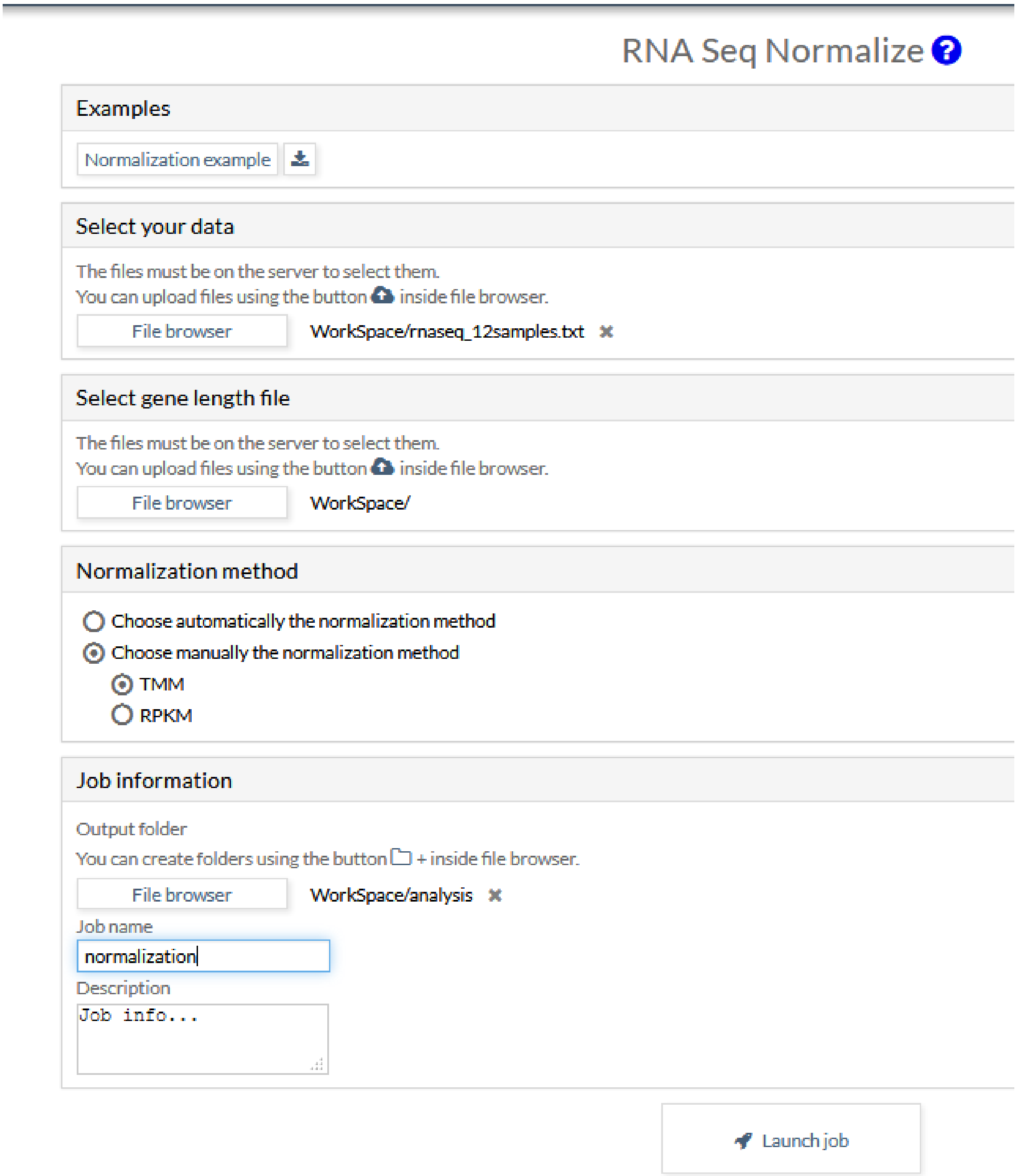
Work plan

1. Open the data file of **rnaseq_12samples.txt** with a spreadsheet and inspect its contents. There will be as many columns as subjects and as many rows as genes.
2. Upload this txt file in Babelomics from the “Upload” menu. We will have to indicate the type of data that we upload: “Data matrix expression”. This link describes the different types of data that we can use in Babelomics: https://github.com/babelomics/babelomics/wiki/Data-types.
3. After loading the data, the first step will be normalization. From “Processing / Normalization NGS: RNA-Seq” we will select our file and choose a standardization method (we will start with TMM). Interesting clue: when the normalization finishes, check out the results and in the “Job information” section, look up the identifier of the “Output folder”. Then we will need it to indicate to Babelomics where are the normalized data.
4. Once the data is already normalized, we are ready to perform the clustering. From “Expression / Unsupervised analysis”, select the data (now it's time to select the previous “output folder” where the normalized data are ready).
5. Next, we select the clustering by samples. We chose a method of clustering and distance (to begin with, those that are by default). We assign a name to the job and run it.
6. Perform a clustering for genes (to begin with, those that are by default). We assign a name to the job and execute it.

Questions

1. Are there groups of samples with a similar transcriptomic profile? How many groups appear?
2. Is there any sample that has an anomalous behavior when comparing with other subjects?

3. Do you think that if we performed a differential expression analysis we would obtain a large number of differentially expressed genes?
4. Any incidence with clustering by genes?



RNA Seq Normalize

Examples

Normalization example

Select your data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/rnaseq_12samples.txt

Select gene length file

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/

Normalization method

○ Choose automatically the normalization method
● Choose manually the normalization method
● TMM
○ RPKM

Job information

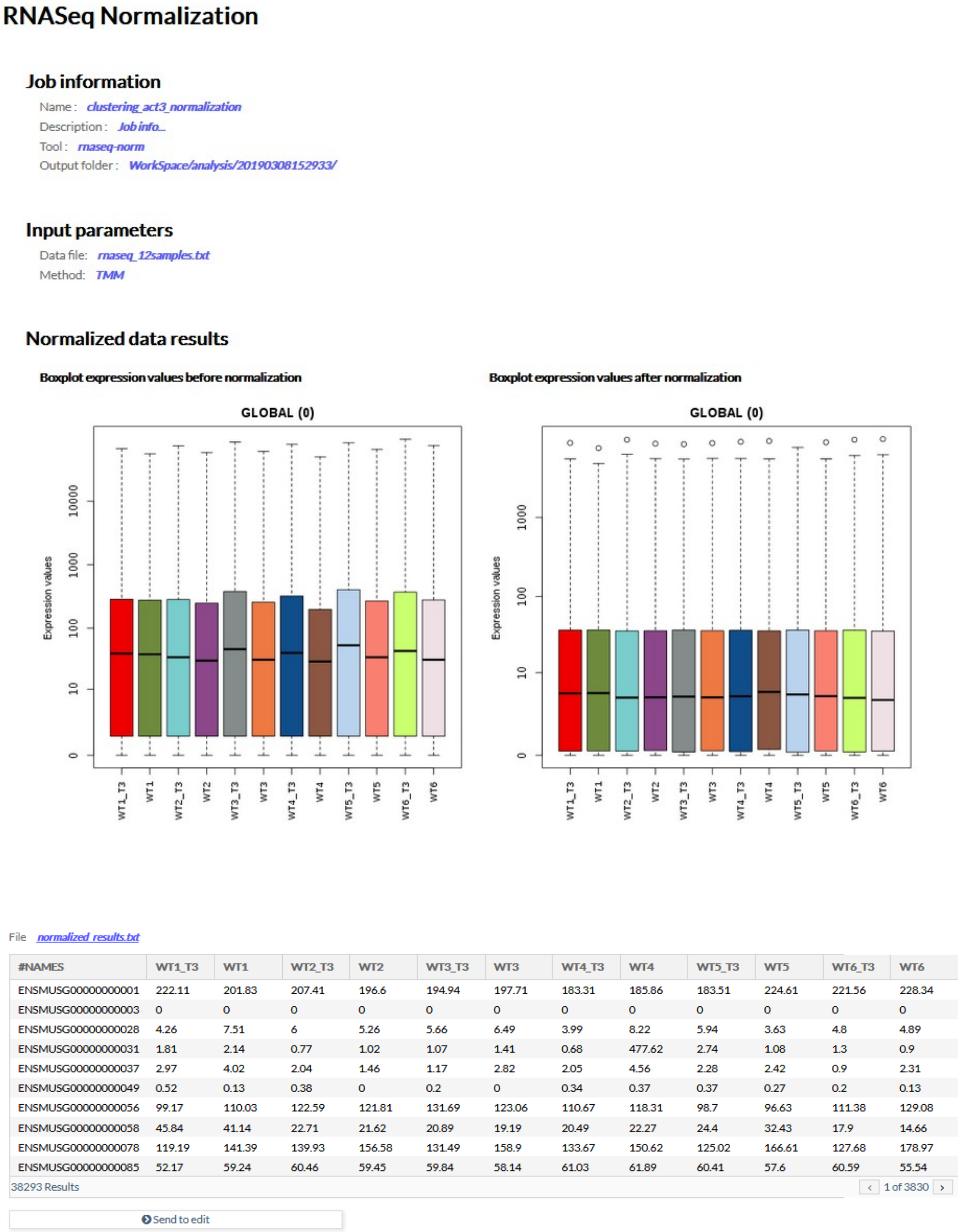
Output folder
You can create folders using the button + inside file browser.

File browser WorkSpace/analysis

Job name
normalization

Description
Job info...

Launch job



RNASeq Normalization

Job information

Name : *clustering_act3_normalization*

Description : *Job info...*

Tool : *rnaseq-norm*

Output folder : *WorkSpace/analysis/20190308152933/*

Input parameters

Data file: *rnaseq_12samples.txt*

Method: *TMM*

Normalized data results

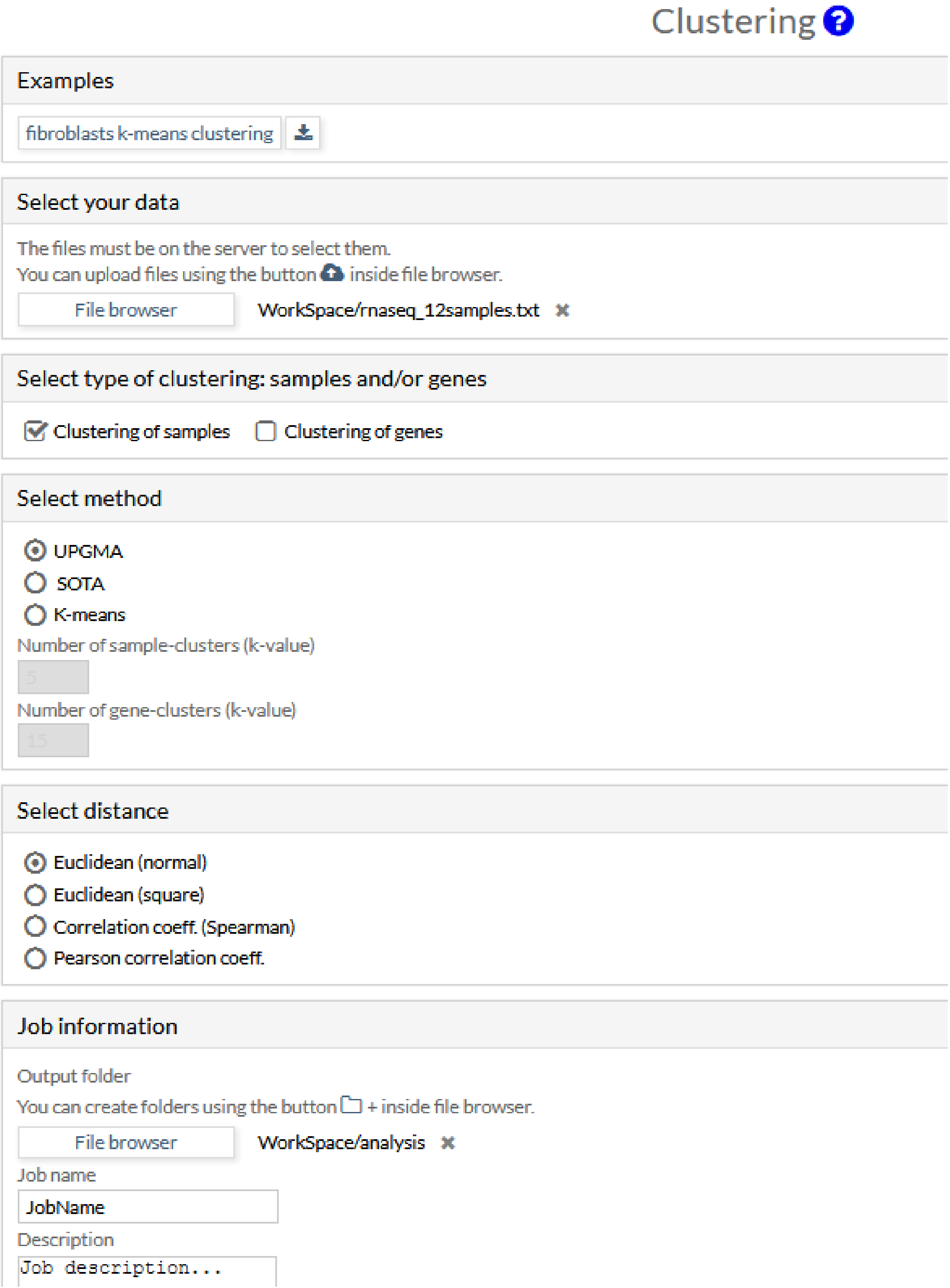
Boxplot expression values before normalization

Boxplot expression values after normalization

File *normalized_results.txt*

#NAMES	WT1_T3	WT1	WT2_T3	WT2	WT3_T3	WT3	WT4_T3	WT4	WT5_T3	WT5	WT6_T3	WT6
ENSMUSG00000000001	222.11	201.83	207.41	196.6	194.94	197.71	183.31	185.86	183.51	224.61	221.56	228.34
ENSMUSG00000000003	0	0	0	0	0	0	0	0	0	0	0	0
ENSMUSG00000000028	4.26	7.51	6	5.26	5.66	6.49	3.99	8.22	5.94	3.63	4.8	4.89
ENSMUSG00000000031	1.81	2.14	0.77	1.02	1.07	1.41	0.68	477.62	2.74	1.08	1.3	0.9
ENSMUSG00000000037	2.97	4.02	2.04	1.46	1.17	2.82	2.05	4.56	2.28	2.42	0.9	2.31
ENSMUSG00000000049	0.52	0.13	0.38	0	0.2	0	0.34	0.37	0.37	0.27	0.2	0.13
ENSMUSG00000000056	99.17	110.03	122.59	121.81	131.69	123.06	110.67	118.31	98.7	96.63	111.38	129.08
ENSMUSG00000000058	45.84	41.14	22.71	21.62	20.89	19.19	20.49	22.27	24.4	32.43	17.9	14.66
ENSMUSG00000000078	119.19	141.39	139.93	156.58	131.49	158.9	133.67	150.62	125.02	166.61	127.68	178.97
ENSMUSG00000000085	52.17	59.24	60.46	59.45	59.84	58.14	61.03	61.89	60.41	57.6	60.59	55.54

38293 Results



Clustering ?

Examples

fibroblasts k-means clustering ⭳

Select your data

The files must be on the server to select them.
You can upload files using the button ☁ inside file browser.

File browser WorkSpace/rnaseq_12samples.txt ✖

Select type of clustering: samples and/or genes

☑ Clustering of samples □ Clustering of genes

Select method

● UPGMA
○ SOTA
○ K-means

Number of sample-clusters (k-value)
5

Number of gene-clusters (k-value)
15

Select distance

● Euclidean (normal)
○ Euclidean (square)
○ Correlation coeff. (Spearman)
○ Pearson correlation coeff.

Job information

Output folder
You can create folders using the button 📁+ inside file browser.

File browser WorkSpace/analysis ✖

Job name
JobName

Description
Job description...

Job information

Name : clustering_act3_upgma_eu

Description : Job description...

Tool : clustering

Output folder : WorkSpace/analysis/20190308153315/

Input parameters

Dataset file name: rnaseq_12samples.txt

Clustering of: samples

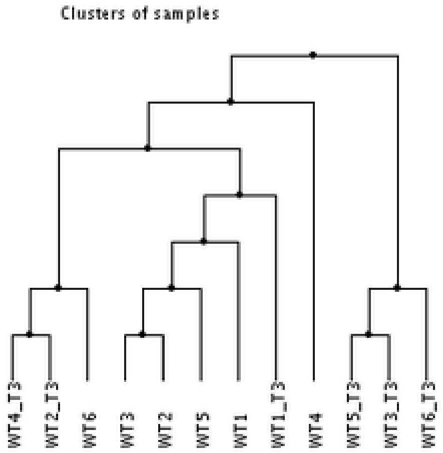
Method: upgma

Distance: euclidean

Clusters in newick format

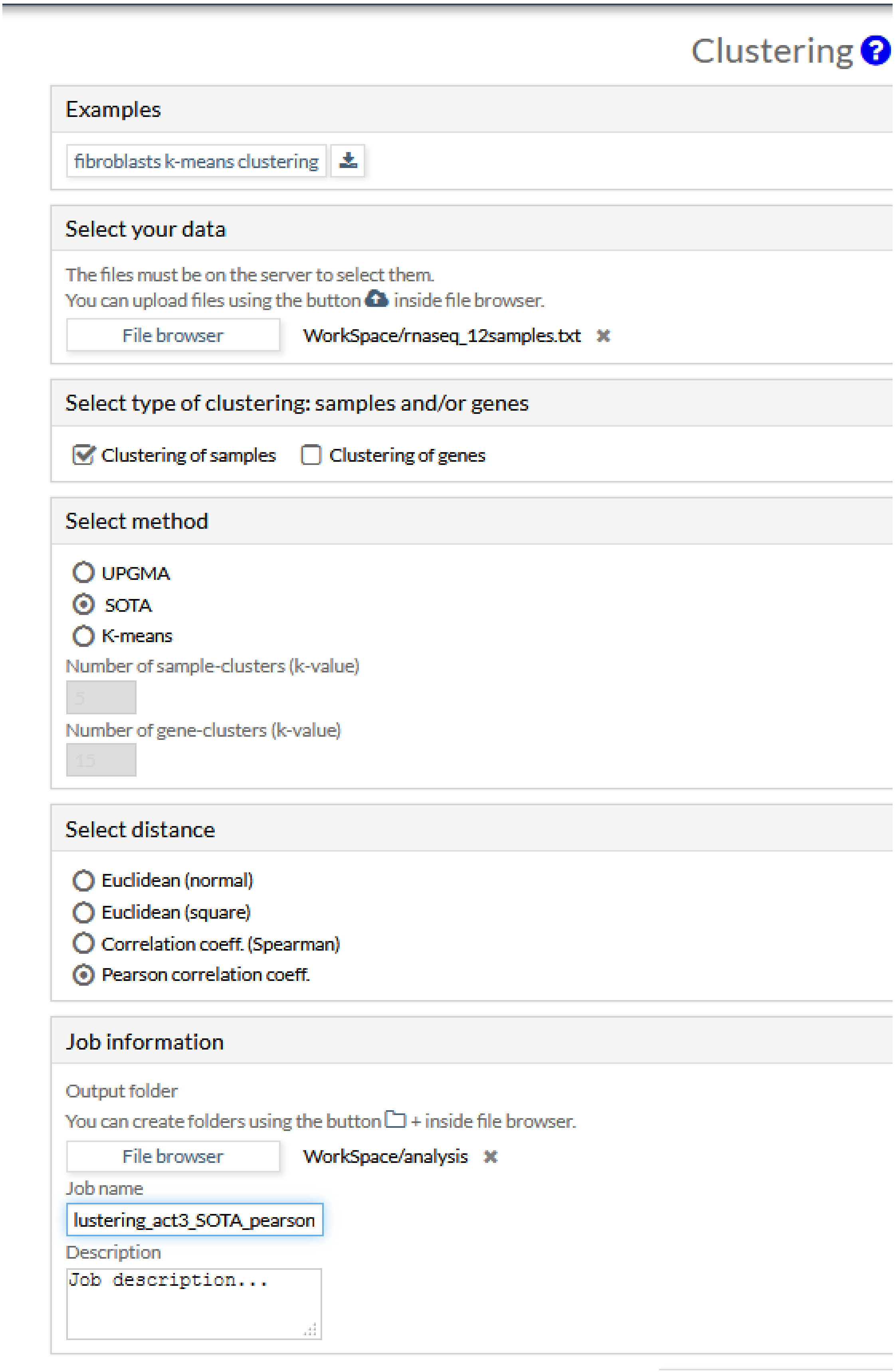
Clusters of samples samples.nw

Cluster images



Warnings

Warning: This release limits the heatmap in tree images to 1000 genes



Clustering ?

Examples

fibroblasts k-means clustering

Select your data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/rnaseq_12samples.txt ✖

Select type of clustering: samples and/or genes

☑ Clustering of samples □ Clustering of genes

Select method

○ UPGMA
● SOTA
○ K-means

Number of sample-clusters (k-value)

5

Number of gene-clusters (k-value)

15

Select distance

○ Euclidean (normal)
○ Euclidean (square)
○ Correlation coeff. (Spearman)
● Pearson correlation coeff.

Job information

Output folder

You can create folders using the button + inside file browser.

File browser WorkSpace/analysis ✖

Job name

lustering_act3_SOTA_pearson

Description

Job description...

Job information

Name : *clustering_act3_SOTA_pearson*

Description : *Job description...*

Tool : *clustering*

Output folder : *WorkSpace/analysis/20190308161640/*

Input parameters

Dataset file name: *rnaseq_12samples.txt*

Clustering of: *samples*

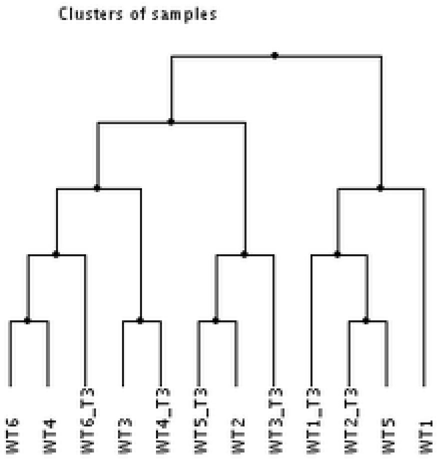
Method: *sota*

Distance: *pearson*

Clusters in newick format

Clusters of samples *samples.nw*

Cluster images



Warnings

Warning: *This release limits the heatmap in tree images to 1000 genes*

C.2. Supervised classification

Predictors are used to assign a new data (expression, proteins, metabolites…) to a specific class (e.g. diseased case or healthy control) based on a rule constructed with a previous dataset containing the classes among which we aim to discriminate. This dataset is usually known as the **training** set. The rationale under this strategy is the following: if the differences between the classes (our macroscopic observations, e.g. cancer versus healthy cases) is a consequence of certain differences an gene level, and these differences can be measured as differences in the level of gene expression, then it is (in theory) possible finding these gene expression differences and use them to assign the class membership for a new array. This is not always easy, but can be aimed. There are different mathematical methods and operative strategies that can be used for this purpose.

In Babelomics, there is an unsupervised classification module to help in the process of building a "good predictor". In this resource:

- We have implemented several widely accepted strategies so as this tool can build up simple, yet powerful predictors, along with a carefully designed cross-validation of the whole process (in order to avoid the widespread problem of "selection bias").
- Babelomics allows combining several classification algorithms with different methods for gene selection.
- Main indicators to assess the quality of prediction: accuracy, MCC, AUC and RMSE.
- More detailed information about methods.

Activities

We have prepared two activities to know how is possible the generation of predictors from Babelomics.

1. Class prediction in acute leukemia.
2. Supervised classification for RNA-Seq data of Lung squamous cell carcinoma.

Here you have more detailed information about *supervised classification module* in Babelomics

Activity 1. Class prediction in acute leukemia

In this example we are going to analyse a dataset from Golub et al. (1999). In that paper they were studying two different types of leukemia (acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) in order to detect differences between them. This dataset have 3051 genes and 38 arrays, 27 of them labeled as ALL and 11 of them as AML.

Using Class prediction we are going to build a predictor to try to distinguish between both classes. In the train file we can see 30 arrays, 21 ALL and 9 AML. The rest, 6 ALL and 2 AML, are in the test file for predicting.

You can find the dataset for this exercise in the following files:

- The first one is the file to train the predictor: datatraingolub.txt.
- The second one will be used to predict the classes (test dataset): datatestgolub.txt.

A. Training

- Train with KNN algorithm. Upload the datafile and select the variable TUMOR. In order to get the exercises fast select 5 repeats of 5-fold cross validation. In this exercise do not select any feature selection method.
- Repeat the exercise but select CFS feature selection method, which one works better? Why? how many genes were selected
- Now try with SVM algorithm with no feature selection method, which one performs better? SVM or KNN
- To finish you can try SVM with CFS feature selection method, how many features were selected? Why it matches KNN with CFS?
- Finally, which is the best combination? Why is SVM doing better along than with CFS?

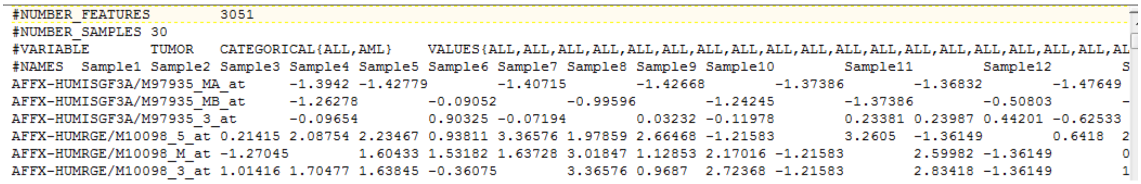
B. Test

- Now we select the option Train and test and select datatraingolub and datatestgolub.
- We can select KNN without feaure method to speed up the exercise.
- In order to check the accuracy of prediction you can see the correct labels for the test file:

```
ALL    ALL      ALL      ALL      ALL      ALL      AML      AML
```

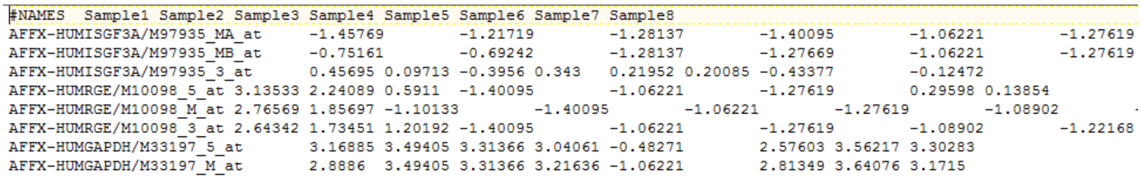
- Are the predictions right? Do you get the same results with SVM?

Train data:

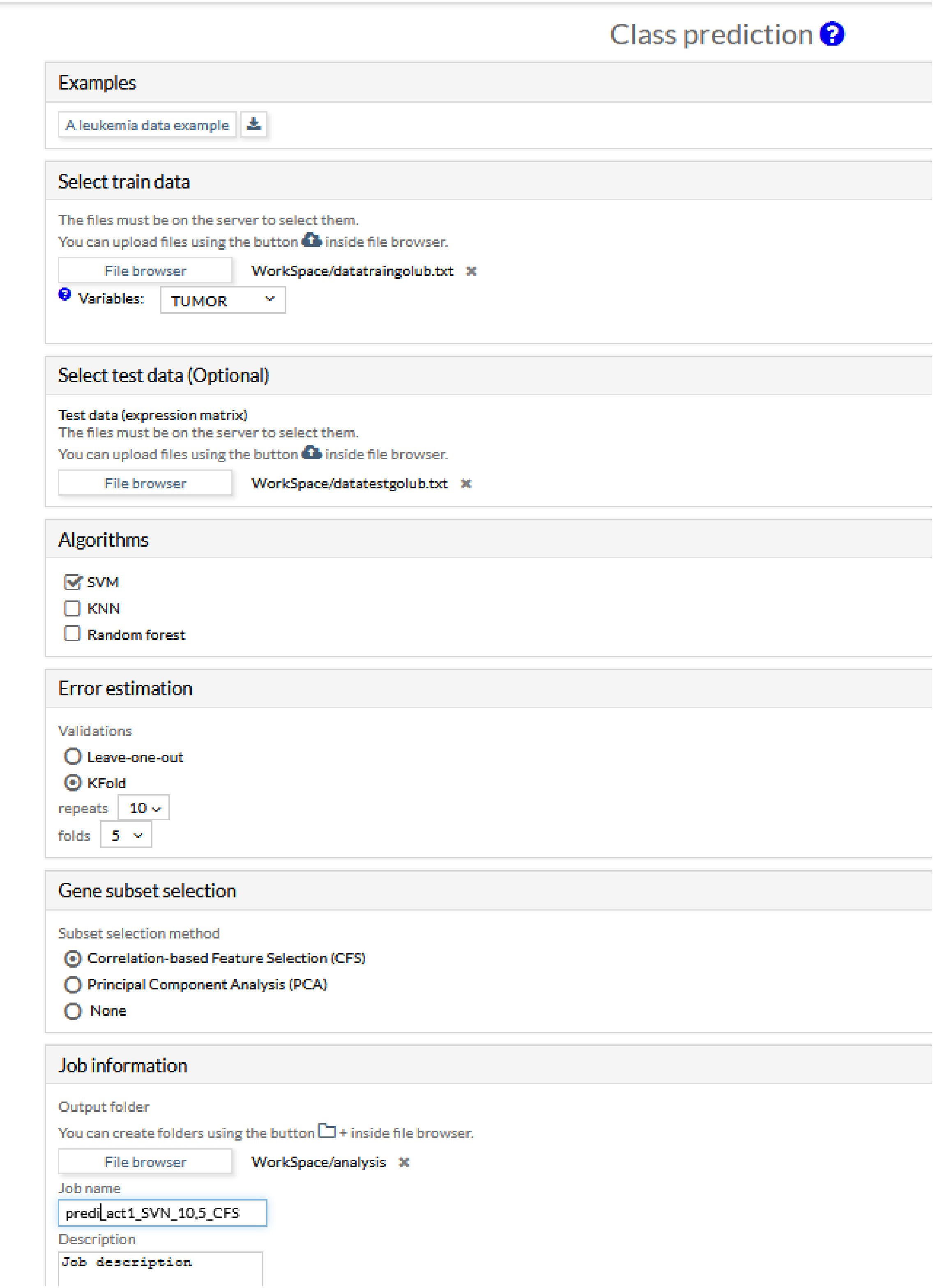


```
#NUMBER_FEATURES        3051
#NUMBER_SAMPLES 30
#VARIABLE       TUMOR   CATEGORICAL{ALL,AML}    VALUES{ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,ALL,AL
#NAMES  Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 Sample7 Sample8 Sample9 Sample10        Sample11        Sample12        S
AFFX-HUMISGF3A/M97935_MA_at     -1.3942 -1.42779        -1.40715        -1.42668        -1.37386        -1.36832        -1.47649
AFFX-HUMISGF3A/M97935_MB_at     -1.26278        -0.09052        -0.99596        -1.24245        -1.37386        -0.50803        -
AFFX-HUMISGF3A/M97935_3_at      -0.09654        0.90325 -0.07194        0.03232 -0.11978        0.23381 0.23987 0.44201 -0.62533
AFFX-HUMRGE/M10098_5_at 0.21415 2.08754 2.23467 0.93811 3.36576 1.97859 2.66468 -1.21583        3.2605  -1.36149        0.6418  2
AFFX-HUMRGE/M10098_M_at -1.27045        1.60433 1.53182 1.63728 3.01847 1.12853 2.17016 -1.21583        2.59982 -1.36149        0
AFFX-HUMRGE/M10098_3_at 1.01416 1.70477 1.63845 -0.36075        3.36576 0.9687  2.72368 -1.21583        2.83418 -1.36149        1
```

Test data



```
#NAMES  Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 Sample7 Sample8
AFFX-HUMISGF3A/M97935_MA_at     -1.45769        -1.21719        -1.28137        -1.40095        -1.06221        -1.27619
AFFX-HUMISGF3A/M97935_MB_at     -0.75161        -0.69242        -1.28137        -1.27669        -1.06221        -1.27619
AFFX-HUMISGF3A/M97935_3_at      0.45695 0.09713 -0.3956 0.343   0.21952 0.20085 -0.43377        -0.12472
AFFX-HUMRGE/M10098_5_at 3.13533 2.24089 0.5911  -1.40095        -1.06221        -1.27619        0.29598 0.13854
AFFX-HUMRGE/M10098_M_at 2.76569 1.85697 -1.10133        -1.40095        -1.06221        -1.27619        -1.08902        -
AFFX-HUMRGE/M10098_3_at 2.64342 1.73451 1.20192 -1.40095        -1.06221        -1.27619        -1.08902        -1.22168
AFFX-HUMGAPDH/M33197_5_at       3.16885 3.49405 3.31366 3.04061 -0.48271        2.57603 3.56217 3.30283
AFFX-HUMGAPDH/M33197_M_at       2.8886  3.49405 3.31366 3.21636 -1.06221        2.81349 3.64076 3.1715
```



Class prediction

Examples

A leukemia data example

Select train data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/datatraingolub.txt ✖

Variables: TUMOR

Select test data (Optional)

Test data (expression matrix)
The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/datatestgolub.txt ✖

Algorithms

☑ SVM
☐ KNN
☐ Random forest

Error estimation

Validations

○ Leave-one-out
● KFold

repeats 10
folds 5

Gene subset selection

Subset selection method

● Correlation-based Feature Selection (CFS)
○ Principal Component Analysis (PCA)
○ None

Job information

Output folder

You can create folders using the button + inside file browser.

File browser WorkSpace/analysis ✖

Job name

prediact1_SVN_10,5_CFS

Description

Job description

Job information

Name : *predictor_act1_SVN_10,5_CFS*

Description : *Job description*

Tool : *class-prediction*

Output folder : *WorkSpace/analysis/20190308153554/*

Train

Summary

Combined results (best 5 per classifier) *best_classifiers_table.txt*

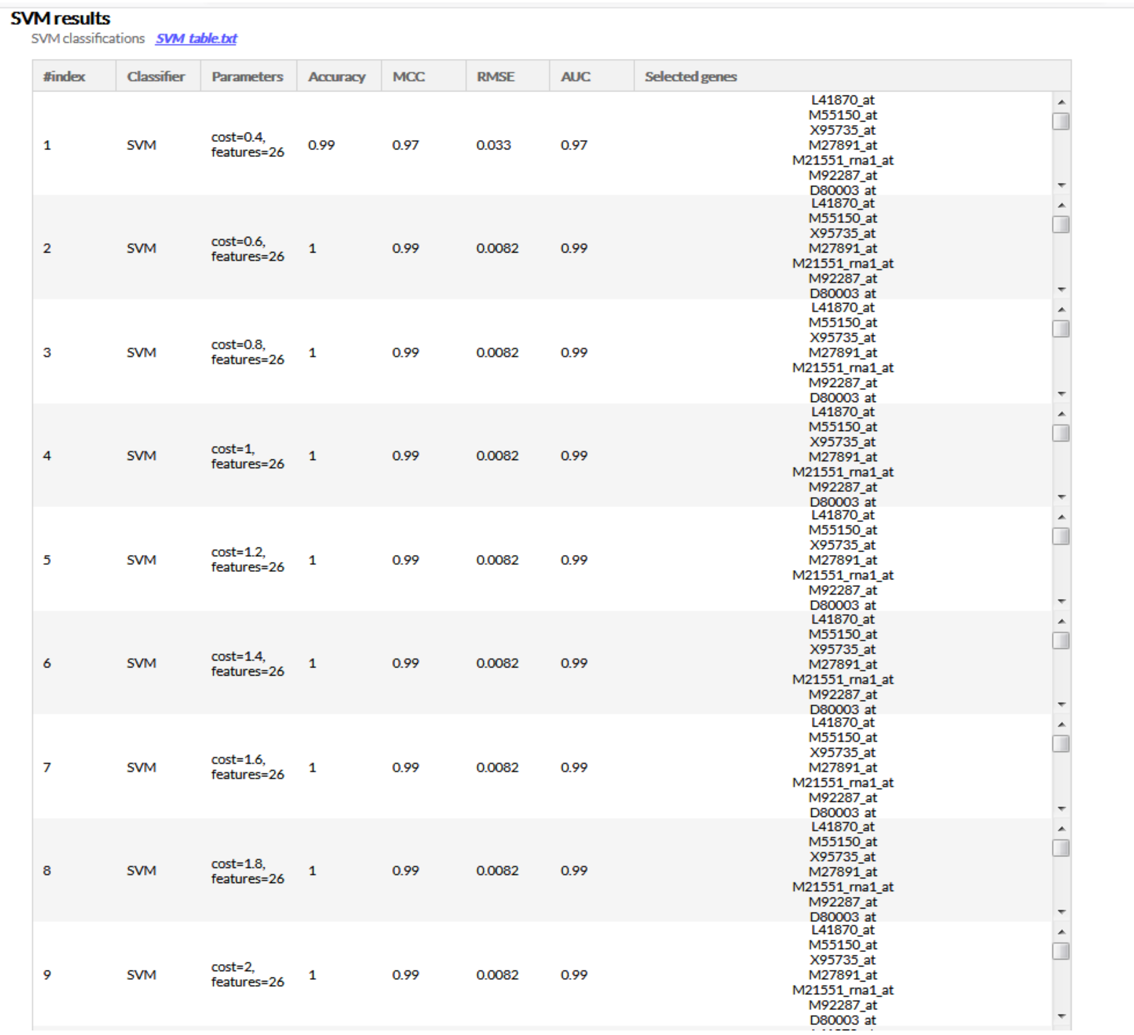
#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
2	SVM	cost=0.6, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
3	SVM	cost=0.8, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
4	SVM	cost=1, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
5	SVM	cost=1.2, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
6	SVM	cost=1.4, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at

5 Results ‹ 1 of 1 ›

Percentage of correct classification per sample/classifier *ratios.html*

#Sample	cost=0.6, features=26	cost=0.8, features=26	cost=1, features=26	cost=1.2, features=26	cost=1.4, features=26
Sample1	100%	100%	100%	100%	100%
Sample2	100%	100%	100%	100%	100%
Sample3	100%	100%	100%	100%	100%
Sample4	100%	100%	100%	100%	100%
Sample5	100%	100%	100%	100%	100%
Sample6	100%	100%	100%	100%	100%
Sample7	100%	100%	100%	100%	100%
Sample8	100%	100%	100%	100%	100%
Sample9	100%	100%	100%	100%	100%
Sample10	100%	100%	100%	100%	100%

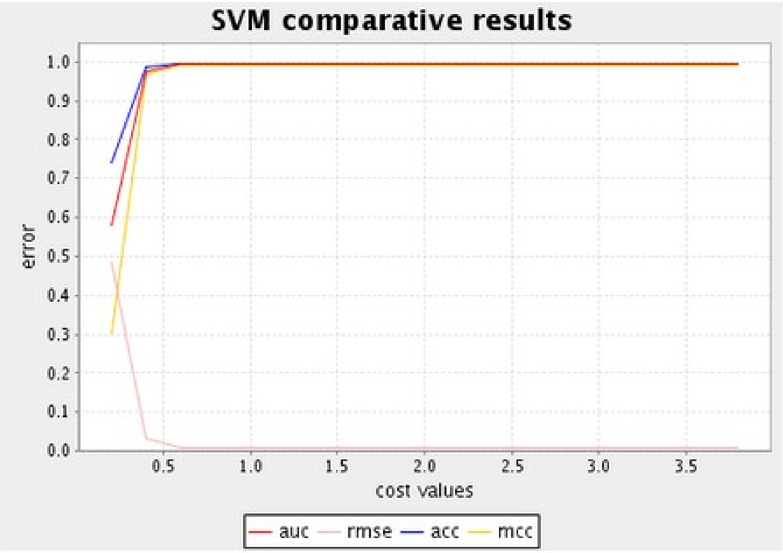
30 Results ‹ 1 of 3 ›



SVM results

SVM classifications *SVM table.txt*

#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
1	SVM	cost=0.4, features=26	0.99	0.97	0.033	0.97	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
2	SVM	cost=0.6, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
3	SVM	cost=0.8, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
4	SVM	cost=1, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
5	SVM	cost=1.2, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
6	SVM	cost=1.4, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
7	SVM	cost=1.6, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
8	SVM	cost=1.8, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at
9	SVM	cost=2, features=26	1	0.99	0.0082	0.99	L41870_at M55150_at X95735_at M27891_at M21551_rna1_at M92287_at D80003_at



Test

Test result

Test result table *test_result.txt*

#Sample_names	SVM cost=0.6, features=26	SVM cost=0.8, features=26	SVM cost=1, features=26	SVM cost=1.2, features=26	SVM cost=1.4, features=26
sample1	ALL	ALL	ALL	ALL	ALL
sample2	ALL	ALL	ALL	ALL	ALL
sample3	ALL	ALL	ALL	ALL	ALL
sample4	ALL	ALL	ALL	ALL	ALL
sample5	ALL	ALL	ALL	ALL	ALL
sample6	ALL	ALL	ALL	ALL	ALL
sample7	ALL	ALL	ALL	ALL	ALL
sample8	ALL	ALL	ALL	ALL	ALL

8 Results ‹ 1 of 1 ›

Activity 2. Supervised classification for RNA-Seq data of Lung squamous cell carcinoma

Data description

RNA-Seq data of Lung squamous cell carcinoma (LUSC) samples taken from The Cancer Genome Atlas (TCGA) data portal.

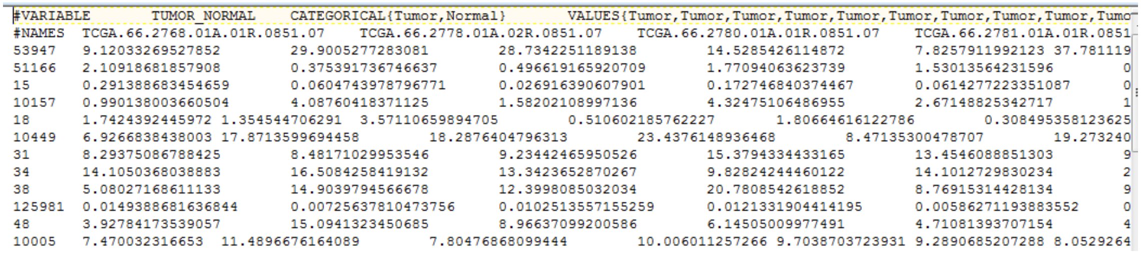
Goals

1. We want to train several classification models in Babelomics.
2. After this step, we are evaluating the best way of classifying our data from a test dataset.

Work plan

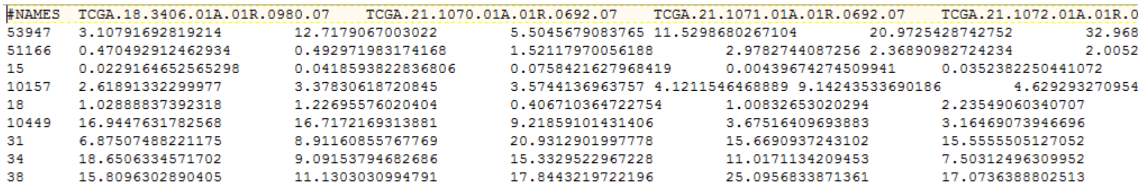
1. Download tca_gene_lusc_train.txt. Contains 11 Normal and 150 Tumor samples.
2. Download tca_gene_lusc_test.txt. Contains 6 Normal and 75 Tumor samples.
3. Upload your files to Babelomics 5.0. Go to section Expression > Class Prediction
4. Try several classification strategies:
   - o Select SVM, KNN and Random Forest
   - o Select Leave-one-out for error estimation
   - o Select Correlation-based Feature Selection (CFS)
5. Download test_result.txt
   - o Which supervised classification method(s) works better?
   - o How many genes were used for the prediction?
   - o Are the selected genes same for all methods?

Train data:

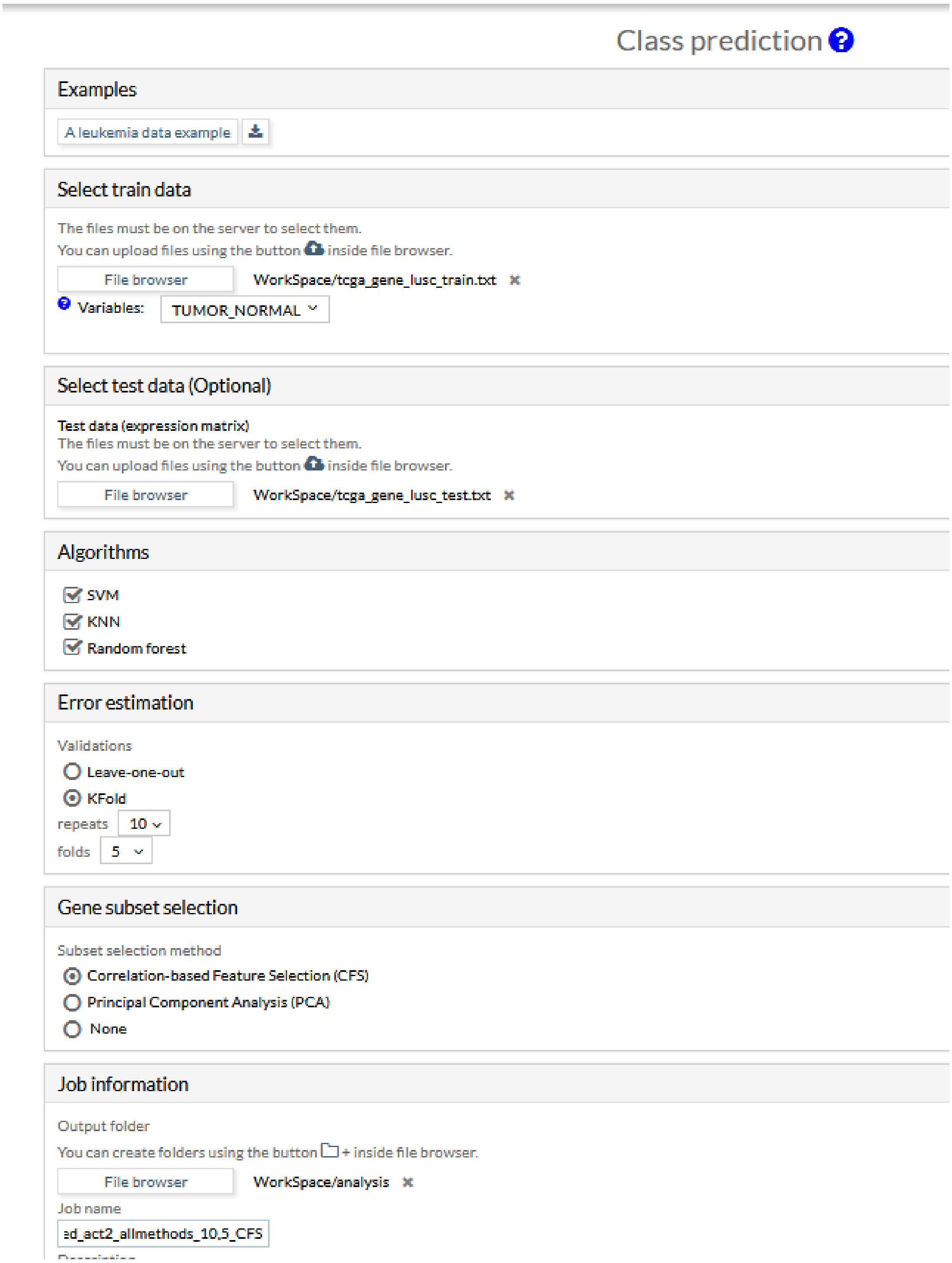


```
#VARIABLE       TUMOR_NORMAL    CATEGORICAL{Tumor,Normal}       VALUES{Tumor,Tumor,Tumor,Tumor,Tumor,Tumor,Tumor,Tumor,Tumor,Tumo
#NAMES  TCGA.66.2768.01A.01R.0851.07    TCGA.66.2778.01A.02R.0851.07    TCGA.66.2780.01A.01R.0851.07    TCGA.66.2781.01A.01R.0851
53947   9.12033269527852        29.9005277283081        28.7342251189138        14.5285426114872        7.8257911992123 37.781119
51166   2.10918681857908        0.375391736746637       0.496619165920709       1.77094063623739        1.53013564231596        0
15      0.291388683454659       0.0604743978796771      0.026916390607901       0.172746840374467       0.0614277223351087      0
10157   0.990138003660504       4.08760418371125        1.58202108997136        4.32475106486955        2.67148825342717        1
18      1.7424392445972 1.354544706291  3.57110659894705        0.510602185762227       1.80664616122786        0.308495358123625
10449   6.9266838438003 17.8713599694458        18.2876404796313        23.4376148936468        8.47135300478707        19.273240
31      8.29375086788425        8.48171029953546        9.23442465950526        15.3794334433165        13.4546088851303        9
34      14.1050368038883        16.5084258419132        13.3423652870267        9.82824244460122        14.1012729830234        2
38      5.08027168611133        14.9039794566678        12.3998085032034        20.7808542618852        8.76915314428134        9
125981  0.0149388681636844      0.00725637810473756     0.0102513557155259      0.0121331904414195      0.00586271193883552     0
48      3.92784173539057        15.0941323450685        8.96637099200586        6.14505009977491        4.71081393707154        4
10005   7.470032316653  11.4896676164089        7.80476868099444        10.006011257266 9.7038703723931 9.2890685207288 8.0529264
```

Test data:



```
#NAMES  TCGA.18.3406.01A.01R.0980.07    TCGA.21.1070.01A.01R.0692.07    TCGA.21.1071.01A.01R.0692.07    TCGA.21.1072.01A.01R.0
53947   3.10791692819214        12.7179067003022        5.5045679083765 11.5298680267104        20.9725428742752        32.968
51166   0.470492912462934       0.492971983174168       1.52117970056188        2.9782744087256 2.3689098272423 4       2.0052
15      0.0229164652565298      0.0418593822836806      0.0758421627968419      0.00439674274509941     0.0352382250441072
10157   2.61891332299977        3.37830618720845        3.5744136963757 4.1211546468889 9.14243533690186        4.629293270954
18      1.02888837392318        1.22695576020404        0.406710364722754       1.00832653020294        2.23549060340707
10449   16.9447631782568        16.7172169313881        9.21859101431406        3.67516409693883        3.16469073946696
31      6.87507488221175        8.91160855767769        20.9312901997778        15.6690937243102        15.5555505127052
34      18.6506334571702        9.09153794682686        15.3329522967228        11.0171134209453        7.50312496309952
38      15.8096302890405        11.1303030994791        17.8443219722196        25.0956833871361        17.0736388802513
```



Class prediction

Examples

A leukemia data example

Select train data

The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/tcga_gene_lusc_train.txt ✖

Variables: TUMOR_NORMAL

Select test data (Optional)

Test data (expression matrix)
The files must be on the server to select them.
You can upload files using the button inside file browser.

File browser WorkSpace/tcga_gene_lusc_test.txt ✖

Algorithms

☑ SVM
☑ KNN
☑ Random forest

Error estimation

Validations
○ Leave-one-out
● KFold

repeats 10
folds 5

Gene subset selection

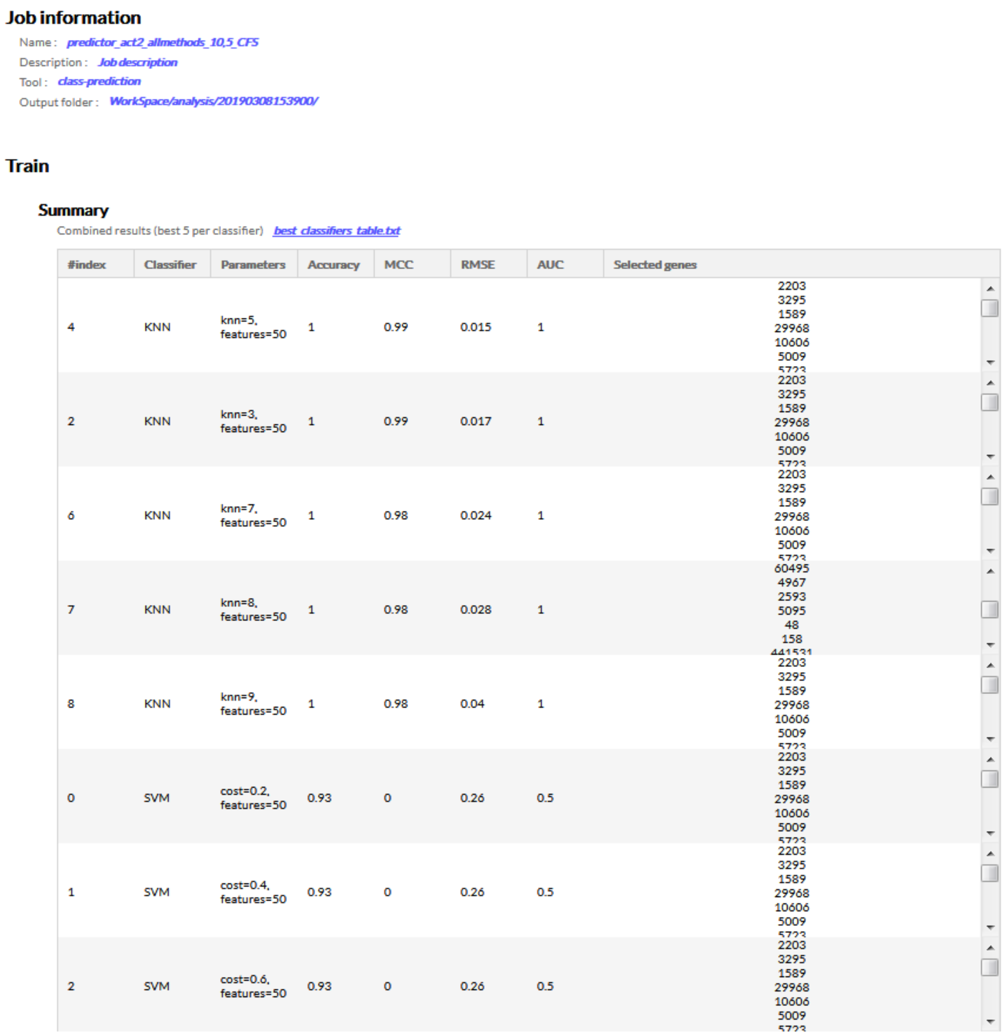
Subset selection method
● Correlation-based Feature Selection (CFS)
○ Principal Component Analysis (PCA)
○ None

Job information

Output folder
You can create folders using the button + inside file browser.

File browser WorkSpace/analysis ✖

Job name
ed_act2_allmethods_10,5_CFS



Job information

Name : *predictor_act2_allmethods_10,5_CFS*

Description : *Job description*

Tool : *class-prediction*

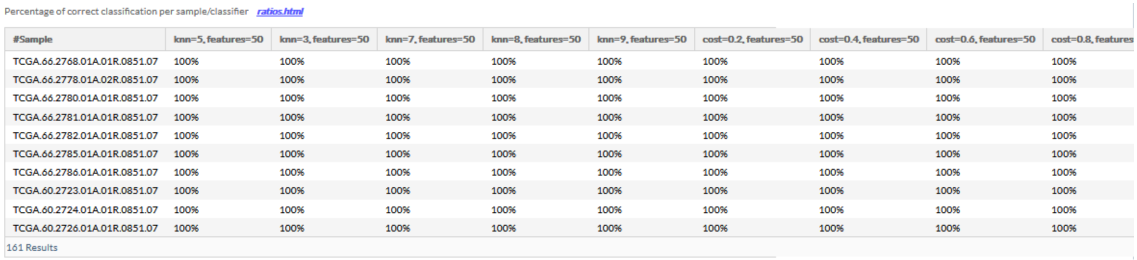
Output folder : *WorkSpace/analysis/20190308153900/*

Train

Summary

Combined results (best 5 per classifier) *best_classifiers_table.txt*

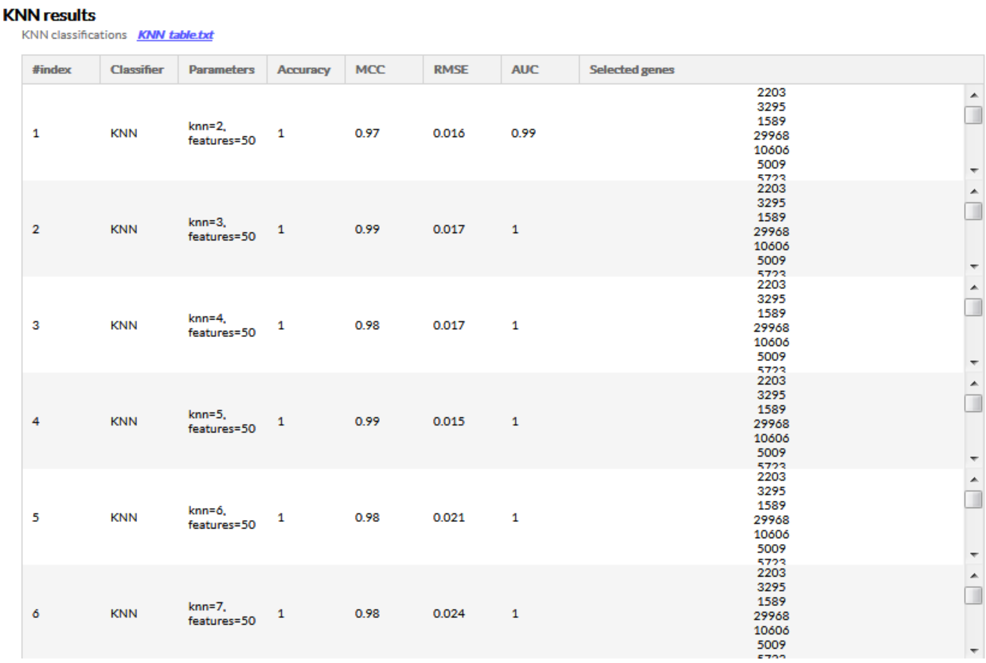
#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
4	KNN	knn=5, features=50	1	0.99	0.015	1	2203 3295 1589 29968 10606 5009 5723
2	KNN	knn=3, features=50	1	0.99	0.017	1	2203 3295 1589 29968 10606 5009 5723
6	KNN	knn=7, features=50	1	0.98	0.024	1	2203 3295 1589 29968 10606 5009 5723
7	KNN	knn=8, features=50	1	0.98	0.028	1	60495 4967 2593 5095 48 158 441531
8	KNN	knn=9, features=50	1	0.98	0.04	1	2203 3295 1589 29968 10606 5009 5723
0	SVM	cost=0.2, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
1	SVM	cost=0.4, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
2	SVM	cost=0.6, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723



Percentage of correct classification per sample/classifier *ratios.html*

#Sample	knn=5, features=50	knn=3, features=50	knn=7, features=50	knn=8, features=50	knn=9, features=50	cost=0.2, features=50	cost=0.4, features=50	cost=0.6, features=50	cost=0.8, features
TCGA.66.2768.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2778.01A.02R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2780.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2781.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2782.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2785.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.66.2786.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.60.2723.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.60.2724.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%
TCGA.60.2726.01A.01R.0851.07	100%	100%	100%	100%	100%	100%	100%	100%	100%

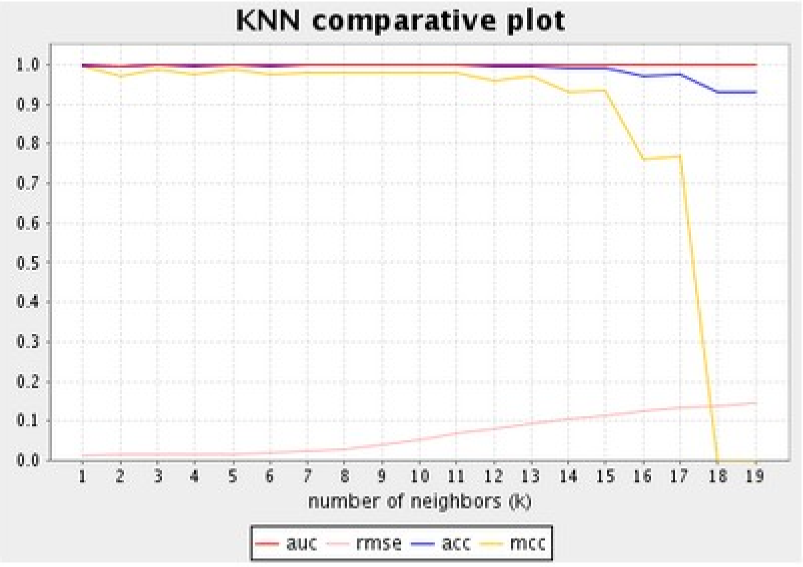
161 Results

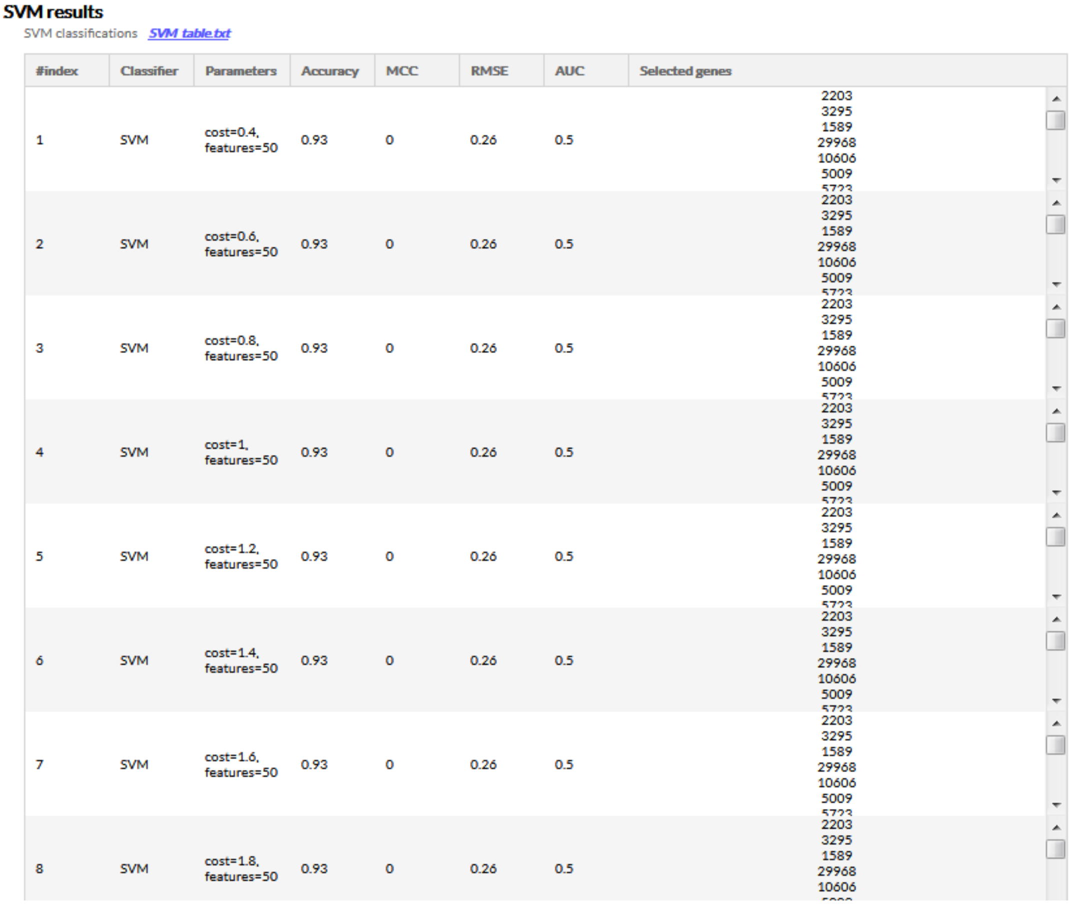


KNN results

KNN classifications *KNN table.txt*

#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
1	KNN	knn=2, features=50	1	0.97	0.016	0.99	2203 3295 1589 29968 10606 5009 5723
2	KNN	knn=3, features=50	1	0.99	0.017	1	2203 3295 1589 29968 10606 5009 5723
3	KNN	knn=4, features=50	1	0.98	0.017	1	2203 3295 1589 29968 10606 5009 5723
4	KNN	knn=5, features=50	1	0.99	0.015	1	2203 3295 1589 29968 10606 5009 5723
5	KNN	knn=6, features=50	1	0.98	0.021	1	2203 3295 1589 29968 10606 5009 5723
6	KNN	knn=7, features=50	1	0.98	0.024	1	2203 3295 1589 29968 10606 5009

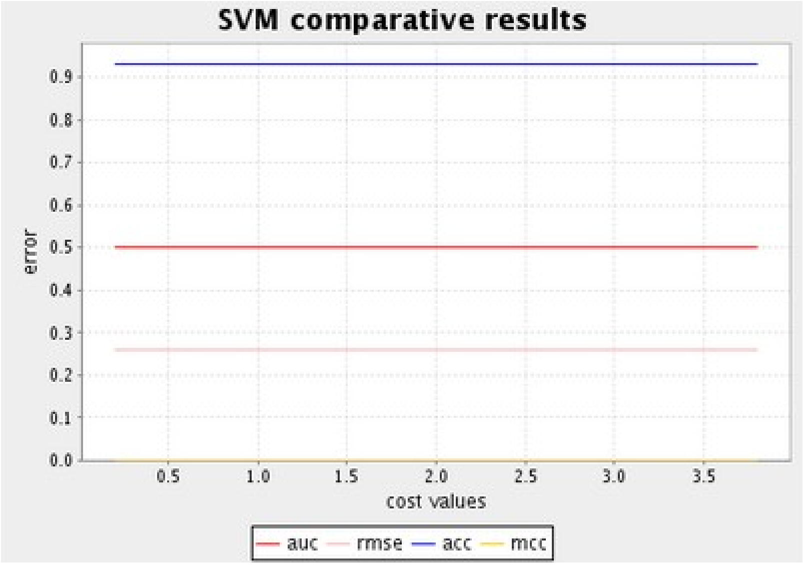


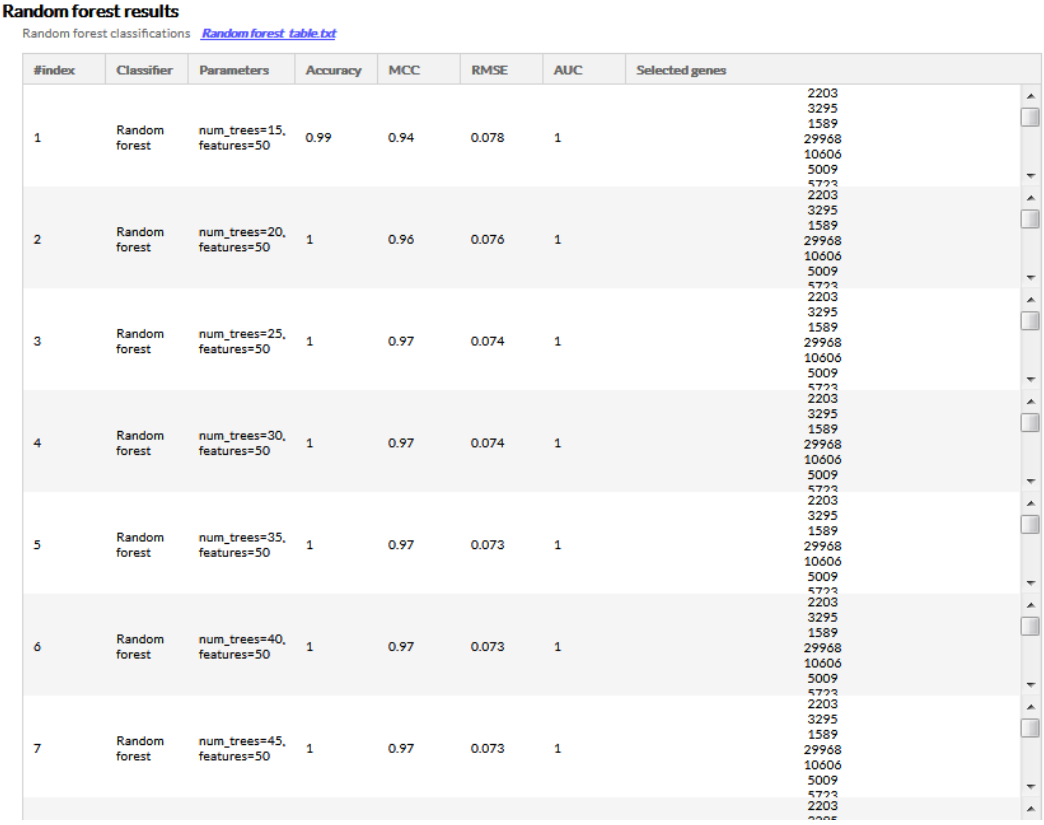


SVM results

SVM classifications *SVM table.txt*

#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
1	SVM	cost=0.4, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
2	SVM	cost=0.6, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
3	SVM	cost=0.8, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
4	SVM	cost=1, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
5	SVM	cost=1.2, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
6	SVM	cost=1.4, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
7	SVM	cost=1.6, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606 5009 5723
8	SVM	cost=1.8, features=50	0.93	0	0.26	0.5	2203 3295 1589 29968 10606

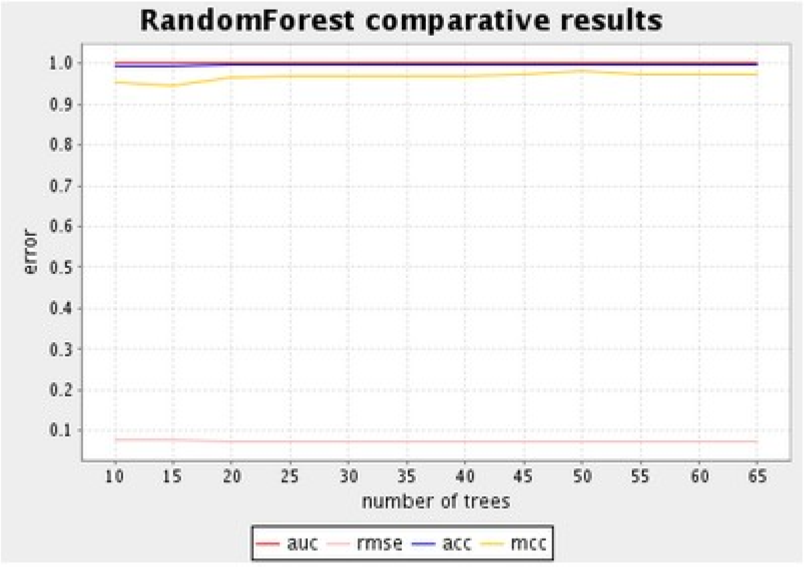


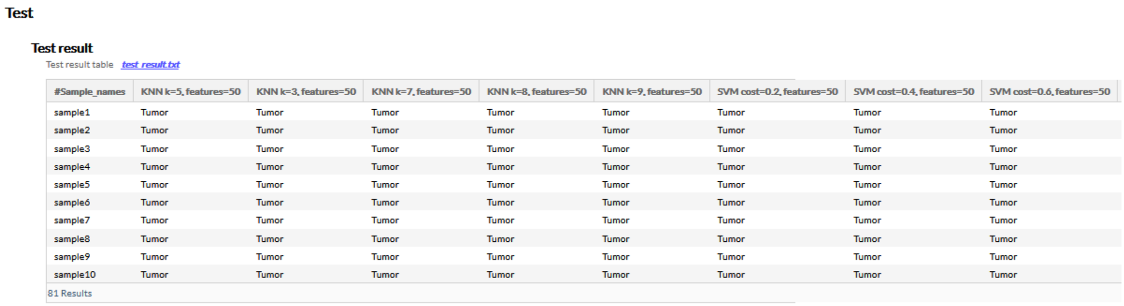


Random forest results

Random forest classifications *Random forest table.txt*

#index	Classifier	Parameters	Accuracy	MCC	RMSE	AUC	Selected genes
1	Random forest	num_trees=15, features=50	0.99	0.94	0.078	1	2203 3295 1589 29968 10606 5009 5723
2	Random forest	num_trees=20, features=50	1	0.96	0.076	1	2203 3295 1589 29968 10606 5009 5723
3	Random forest	num_trees=25, features=50	1	0.97	0.074	1	2203 3295 1589 29968 10606 5009 5723
4	Random forest	num_trees=30, features=50	1	0.97	0.074	1	2203 3295 1589 29968 10606 5009 5723
5	Random forest	num_trees=35, features=50	1	0.97	0.073	1	2203 3295 1589 29968 10606 5009 5723
6	Random forest	num_trees=40, features=50	1	0.97	0.073	1	2203 3295 1589 29968 10606 5009 5723
7	Random forest	num_trees=45, features=50	1	0.97	0.073	1	2203 3295 1589 29968 10606 5009 5723
							2203





Test

Test result

Test result table *test_result.txt*

#Sample_names	KNN k=5, features=50	KNN k=3, features=50	KNN k=7, features=50	KNN k=8, features=50	KNN k=9, features=50	SVM cost=0.2, features=50	SVM cost=0.4, features=50	SVM cost=0.6, features=50
sample1	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample2	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample3	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample4	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample5	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample6	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample7	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample8	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample9	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor
sample10	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor	Tumor

81 Results