BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–6

A Pipeline for Lung Adenocarcinoma tumour (LUAD) prediction using Apache Flink's machine learning and graph library

Sascha Johannes [1]

[1]Freie Universität Berlin, Berlin Germany

ABSTRACT

This paper describes the result of the final assignment of the course "BIG DATA IN LIFE SCIENCE" of the Freie Universitt Berlin. The task of this assignment is to produce a pipeline, which is able to distinguish between two cancer cohorts. The pipeline is developed using the programming language Scala and the streaming dataflow engine Flink[1]. Flink offers a batch processing application programming interface (API) which includes a machine learning and a graph API (called Flink ML and Gelly). The pipeline is trained using RNASeq and miRNASeq data from The Cancer Genome Atlas[2] (TCGA).

Availability:

The pipeline is available through an open git repository, see https://github.com/saschajohannes/flink-luad-pipeline

Contact:

sascha.johannes@fu-berlin.de

1 INTRODUCTION

Non-small cell lung cancer (NSCLC) is the most common type of lung cancer, more than 85% of lung cancer is represented by this subtype. NSCLC divides into three subtypes: adenocarcinomas, squamous cell carcinomas and large cell carcinomas. Lung adenocarcinoma (LUAD) is the most common form of NSCLC, more than 50% of NSCLC are LUAD tumours (see figure 1 for complete lung cancer hierarchy). LUAD often begins in the outer part of the lung and is often overseen in its early stages. Early symptoms are breath shortness in a mild form and/ or pain in the shoulder and the breast. Later stages are accompanied by chronic caught, coughing blood and breath shortness (non-mild form). As in all forms of lung cancer, smoking as well as radiation are the most common causes for LUAD. The survival rate ranges from 49% in early stages to only 1% in later stages. Typical treatments are chemo- and radiation-therapies and surgeries. LUAD is as well as other (lung) cancer types divided into four diagnostic stages:

- Stage 1: The cancer has not infiltrated any lymph node (survival rate: 45-49%)
- Stage 2: The cancer has infiltrated lymph nodes and/ or is located in the main bronchus (survival rate: 31%)
- Stage 3: The cancer has infiltrated tissue outside the lung (survival rate: 5-14%, survival time (with treatment): ∼13 months)
- Stage 4: The cancer has metastasized (survival rate: 1%, survival time (with treatment): ∼8 months)

Platform	# TN	# NT	# samples
Agilent G4502A	0	32	32
Human Methylation 27k	126	24	150
Human Methylation 450k	460	32	492
Genome-Wide			
Human SNP Array 6.0	520	600	1120
HiSeq RNASeq	125	37	162
GA miRNASeq	0	63	63
HiSeq miRNASeq	452	46	498
MDA RPPA	365	0	365
HiSeq RNASeqV2	517	59	576
HiSeq DNASeqC	120	129	249
HiSeq WGBS	5	1	6
Samples	587	698	1258

Table 1. Platforms of the LUAD TCGA dataset, including the number of tumorous samples (TN), non tumorous samples (NT) and the complete number of samples

The Cancer Genome Atlas offers data for Lung squamous cell carcinoma (LUSC) and LUAD. The produced pipeline uses LUAD data. There are different platforms available including RNASeq, miRNASeq, methylation, CNV, Mutations and others. The LUAD dataset contains 1258 samples, 587 tumorous and 698 non-tumorous samples. Each sample was analyzed using a variate of platforms, but not all samples are analyzed with all platforms (see table 1).

To analyze batch data like the TCGA datasets, apache Flink offers a graph and machine learning api, which includes a variety of different methods to get the most of these data sets.

[1] https://flink.apache.org

[2] https://tcga-data.nci.nih.gov

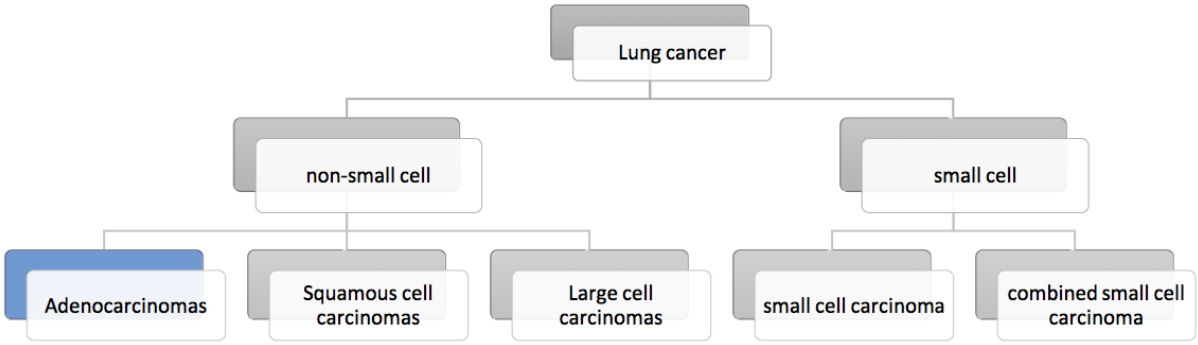

Fig. 1. Hierarchy of lung cancer. Lung cancer divides into two main subtypes: small cell lung cancer and non-small lung cancer. Small cell lung cancer also divides into different subtypes: adenocarcinomas, squamous cell carcinomas and large cell carcinomas. Non-small cell lung cancer is divided into two main subtypes: small cell carcinoma and combined small cell carcinoma. The pipeline which was produced uses adenocarcinomas lung cancer data.

2 DATA

2.1 The Genome Cancer Atlas

Since the data from the TCGA website is inhomogeneous according to their platform coverage, only Illumina HiSeq miRNASeq and Illumina HiSeq RNASeq data is chosen to be used. Both dataset have a large overlap. 58 Samples where randomly chosen from the TCGA website, each samples contains at least miRNASeq or RNASeq data. 35 of them are represented by miRNASeq and RNASeq data. Additional two samples from the TCGA LUSC data are chose (one tumorous and one non-tumorous) and two samples of the cancer type Head and Neck squamous cell carcinoma (HNSC, one tumorous and one non-tumorous). All samples which are not from the LUAD dataset are only used for prediction and not for the training process. 26 samples (24 LUAD, 1 LUSC, 1 HNSC) are labeled as tumorous, 36 (34 LUAD, 1 LUSC, 1 HNSC) are labeled as non-tumorous. The dataset is split into two parts one for the training and one which should be predicted. The training dataset contains $\sim 66\%$ of the samples, 15 tumorous and 25 non-tumorous samples. The sampleset which should be predicted contains 11 tumorous and 11 non tumorous samples. Both sets contains complete sets (samples with miRNA and RNA data) and incomplete samples (samples where only miRNA or RNA data is present). The training set contains 30 complete and 10 incomplete samples (7 RNA and 3 miRNA samples). The prediction set contains 9 complete samples and 13 incomplete samples (7 RNA and 6 miRNA samples). For a complete sample list see table 2.

3 METHODS

The pipeline is developed using the Scala programming language

3.1 Alternating least square

The Alternating least square algorithm (ALS) is a matrix completion algorithm which is based on a recommendation principle. The ALS factorizes a given matrix into two factors such that the following holds.

$$R \approx U^T V$$

Where R is the given matrix and calculated U and V are the factors. To find the matrix U and V the following problem is solved:

$$\arg\min_{U,V} \sum_{\{i,j|r_{ij}\neq 0\}} \left(r_i, j - u_i^T v_j\right)^2 + \lambda \left(\sum_i n_{u_i} ||u_i||^2 \sum_j n_{v_j} ||v_j||^2 \right)$$

Where $r_{i,j}$, u_i and v_i represents the entry of the matrix R, U and V. λ is the regularization factor, n_{u_i} the number of elements which are present in column i and n_{v_j} the number elements which are present in row j. The implementation allows some parametrization to tune the performance and accuracy of the completion (Apache Flink [2015a]):

- NumFactors - Value: 10, number of latent factors, dimension of U and V
- Lambda - Value: 0.9, the regularization factor
- Iterations - Value: 10, maximum number of iterations
- Blocks - Value: 100, number of block into which the which U and V are grouped
- Seed - Value: 42, the seed for the randomly generated initial U and V

3.2 Support Vector Machine

Flink implements the Support Vector Machine (SVM) with soft-margins using communication-efficient distributed dual coordinate ascent algorithm with hinge-loss function. The algorithm solves the given optimization problem:

$$\min_{w\in\mathbb{R}^d} \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} l_i \left(w^T x_i\right)$$

where w is the weight vector, λ the regularization constant, $x_i \in \mathbb{R}$ the data points and l_i the convex loss function. Where l_i is defined as:

$$l_i = \left(0, 1 - y_i w^T x_i\right)$$

Where $y_i \in \mathbb{R}$ are the corresponding labels. The implementation allows various parametrization, for each parameter the default value is used (Apache Flink [2015c]):

- Blocks - Value: NONE, means that the input is split into a specific number of splits which could be processed parallel. A value of NONE means that the number of blocks is equal to the degrees of parallelism of the used machine.
- Iterations - Value: 10, maximum number of outer iterations.
- LocalIterations - Value: 10, number of chosen data-points in each iterations.
- Regularization - Value: 1.0, the value of λ, effects the SVM margin.
- Stepsize - Value: 1.0, tunes the stability of the algorithm, effects the update step size of the weight vector
- ThresholdValue - Value: 0.0, the value where positive and negative predictions are distinguished regarding the decision function
- OutputDecisionFunction - Value: false, show labels as out (false) or the distance to the hyperplane (true)

- Seed - Value: random number, initial value for the random number generator which determines the chosen data-points

3.3 Co-expression Network

In order to reduce the number of probes which are used for the SVM, a co-expression network is build. The network is build using the Pearson Correlation coefficient. Each probe-set is correlate with each probe-set. With the resulting correlation coefficients a graph is build, where each probe-set represents a vertex and and a vertex is connected to another vertex if the absolute value of the corresponding correlation coefficient is larger than a given threshold. Afterwards a connected component algorithm is applied on the resulting graph.

3.3.1 Pearson Correlation Coefficient The Pearson Correlation Coefficient (PCC) measures the linear correlation of two variables. The resulting coefficient is computed using the following formula:

$$r = \frac{n\left(\sum xy\right) - \left(\sum x\right)\left(\sum y\right)}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$

Where x and y represent the variables which are correlated, n refers to the size of both variables. The coefficient ranges from -1.0 (negative correlation) over 0.0 (no correlation) to $+1.0$ (positive correlation).

3.3.2 Connected Components The connected Component algorithm finds vertices which are connected inside a directed graph and labels them with an component id. There is only on parametrization possible (source code taken from Apache Flink [2015b]):

- maxIterations - Value: 100, number of iterations to find attached components to a vertex

Sample ID	RNA	miRNA	Tumorous	Prediction	Sample ID	RNA	miRNA	Tumorous	Prediction
TCGA-49-6744-11	×	×	×	×	TCGA-91-6835-11	×	×	×	×
TCGA-44-6776-11	×	×	×	×	TCGA-44-6148-11	×		×	×
TCGA-38-4632-11	×		×	×	TCGA-50-5931-11	×		×	×
TCGA-50-5932-11	×	×	×		TCGA-50-5933-11	×	×	×	
TCGA-49-6745-11	×	×	×		TCGA-44-6778-11	×	×	×	
TCGA-44-6144-11	×	×	×		TCGA-50-5930-11	×	×	×	
TCGA-49-6742-11	×	×	×		TCGA-44-6777-11	×	×	×	
TCGA-49-6743-11	×	×	×		TCGA-91-6836-11	×	×	×	
TCGA-50-5935-11	×		×		TCGA-44-5645-11	×		×	
TCGA-44-6145-11	×		×		TCGA-44-6146-11	×		×	
TCGA-44-6147-11	×		×		TCGA-67-6217-01	×	×		×
TCGA-73-4676-01	×	×		×	TCGA-75-5122-01	×	×		×
TCGA-44-2665-01	×			×	TCGA-44-3918-01	×			×
TCGA-44-2668-01	×			×	TCGA-44-5645-01	×	×		
TCGA-44-6146-01	×	×			TCGA-44-6147-01	×	×		
TCGA-44-6148-01	×	×			TCGA-44-6775-01	×	×		
TCGA-91-6840-01	×	×			TCGA-05-5429-01	×	×		
TCGA-05-5715-01	×	×			TCGA-50-5049-01	×	×		
TCGA-64-5774-01	×	×			TCGA-64-5778-01	×	×		
TCGA-64-5781-01	×	×			TCGA-49-4488-01	×	×		
TCGA-50-5931-01	×	×			TCGA-50-5932-01	×	×		
TCGA-50-5941-01	×	×			TCGA-50-5944-01	×	×		
TCGA-50-6593-01	×	×			TCGA-55-6543-01	×	×		
TCGA-67-6216-01	×	×			TCGA-44-4112-01	×			
TCGA-44-2656-01	×				TCGA-44-2655-11		×	×	×
TCGA-44-2665-11		×	×	×	TCGA-44-2668-11		×	×	×
TCGA-38-6178-01		×		×	TCGA-44-6145-01		×		×
TCGA-50-5933-01		×		×	TCGA-67-6215-01		×		
TCGA-50-5939-01		×			TCGA-05-4384-01		×		
LUSC									
TCGA-18-3417-01	×	×		×	TCGA-22-4593-11	×		×	×
HNSC									
TCGA-CV-5444-01	×	×		×	TCGA-CV-6936-11	×	×	×	×

Table 2. List of used TCGA samples each. For each sample it is shown if there is RNA and/ or miRNA data is present. Each sample is marked as tumorous or non-tumorous and if the sample is used for the training or prediction set.

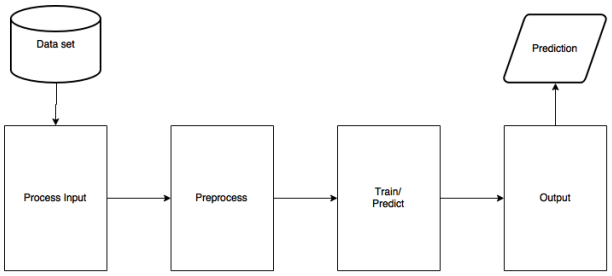

Fig. 2. General workflow of the pipeline. The first step includes the processing of the input file,

```
1  #define a sample name
   def sample <sample-name>
3
   #define a sample to predict
5  def predictive <sample-name>

7  #define sample as tumorous, default: non-tumorous
   diagnosis <sample-name> TN
9
   #define a sample type
11 def sample-type <sample-type>

13 #add a file for a sample-type to a sample
   <sample-type> <sample-name> <file-path>
15
   #define output
17 def output <output-file>

19 #define threshold for correlation
   def pc-threshold <threshold|none>
```

Fig. 3. Possible input data for the input file of the pipeline, line 2: defining a sample with its id, line 5: define a which should be predicted, line 8: set a diagnosis (e.g. tumorous) for a sample, line 11: define a sample type (e.g. rna or mirna), line 14: set a file for a given sample and a given sample type, line 17: define the output file, where all predictions are written (if not given, the output will be printed to the console), line 20: define a threshold for the Pearson correlation coefficient filter step

4 PIPELINE

The given pipeline contains five modules: Input Processing, Preprocessing, Training/ Prediction, Postprocessing and Output (forgeneral workflow see figure 2).

4.1 Input

The Pipeline requires only one file as input. these input file contains all required definitions. The definitions includes all samples, its tumorous states and all different sample-type (e.q. mirna, rna and others). Each sample requires at least a given file for one sample type (for file format see figure 3). The input module reads all defined files and creates one large data matrix (in sparse format). Additional all other given options like tumorous state and output file are set. Each given file is read as probe-name-value association, where each probe-name is converted to an id, which corresponds to the data-matrix column, as well as each sample name which corresponds to a row. If more than one sample-type is specified, the data is simply appended to the data matrix column.

4.2 PreProcessing

The PreProcessing module contains two main steps, a Alternating Least Square based matrix completion and a co-expression network based filter step. All indices which are missing are collected and the corresponding values are predicted using the ALS algorithm. The ALS is trained using the available matrix entries. The second step is only applied if a threshold for the Pearson correlation is available. First a co-expression network is build using the Pearson correlation coefficient. All correlation coefficients are calculated and then filtered using the given threshold such that all remaining values (absolute) are greater than given threshold. In the next step a graph is build using the correlation values, each vertex is represented by a probe-set and each edge is represented by a absolute correlation coefficient large than the threshold. Afterwards a connected components analysis is applied on the graph. For each determined component one node is extracted. the resulting node list is used as a filter such that the data-matrix only contains values which are associated with corresponding nodes in the list (see figure 4 for workflow).

4.3 Training/ Prediction

The preprocessed data-matrix is split into two parts, the first part contains all values which are associated with samples for training. The second part contains only samples where the tumorous state is unknown and therefore should be predicted. The first split is used to train the SVM. Foreach sample in the second split the tumorous state is predicted using the trained SVM.

4.4 Output

After the prediction all predicted samples (represented as id) are converted back their corresponding sample name. the result is then written to an file if an output path is given or printed to the console. The prediction is given as an double value, where -1.0 represents a negative prediction (non-tumorous) and 1.0 represents a positive prediction (tumorous).

5 RESULT

Using the described data set the prediction results in 7 of 22 correct and 15 of 22 wrong predictions using no Pearson correlation based filter step. Using a threshold of 0.8 the pipeline is able to predict 9 of 22 sample correct. One of four samples which does not belong to the training diseases type are predicted correct using the method without correlation filter. Using a correlation filter of 0.8 two sample are predicted correct. See table 3 for complete prediction result.

6 DISCUSSION

In order to evaluate the prediction results of the pipeline, the accuracy, precision, specificity, sensitivity and the negative predictive value (NPV) is calculated. These values are calculated

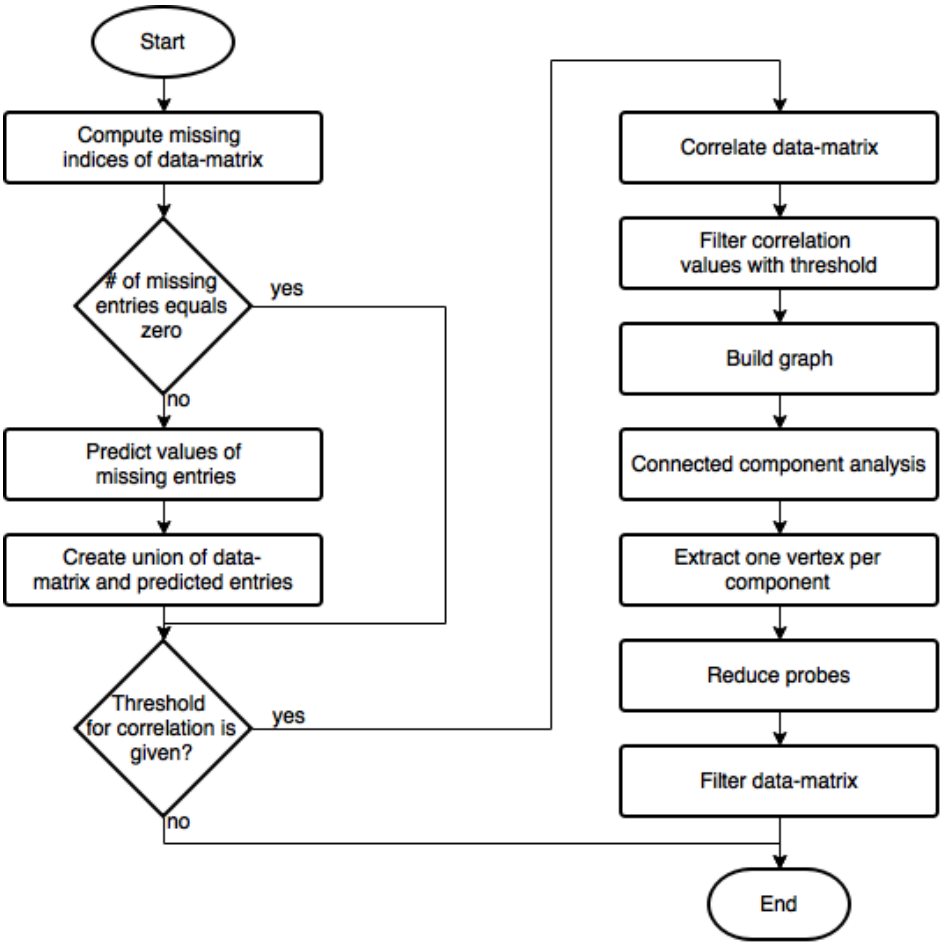

Fig. 4. Workflow of the PreProcessing module, first all missing values of the data-matrix are predicted using the Alternating Least Square method. Afterwards an optional filter step is done. This step includes the building of an co-expression network using the Pearson correlation coefficient and a connected component analysis. Each component is then represented in the filtered data-matrix by only one probe-set.

according to (Thusberg *et al.* [2011]):

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$\text{Precision} = \frac{t_p}{t_p + f_p}$$

$$\text{Specificity} = \frac{t_n}{f_p + t_n}$$

$$\text{Sensitivity} = \frac{t_p}{t_p + f_n}$$

$$\text{NPV} = \frac{t_n}{t_n + f_n}$$

Comparison of these values between both versions (with and without filter process) shows an increase in accuracy (+9%), an increase in precision (+8%) an increase in sensitivity (+18%) and an increase in NPV (+8%). The specificity remains the same. For concrete values see table 4. A possible approach to optimize the accuracy could is to increase the training data.

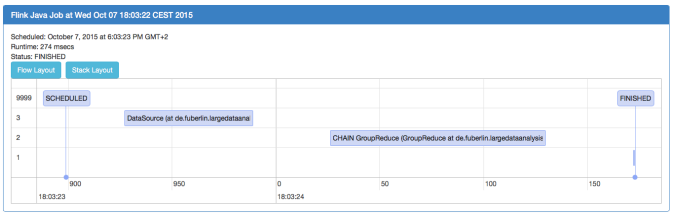

Fig. 5. Result of a ”successful” run on a Flink cluster

6.1 Run it on a cluster

The current pipeline is unable to run on a cluster, it either throw an exception saying that it is unable to connect to the hdfs. Or it runs only partly and finishes after a DataSink(org.apache.flink.api.java.io.DiscardingOutputFormat, see figure 5).

Sample ID	Threshold 0.8	No filter	Expected
TCGA-38-4632-11	1.0	1.0	1.0
TCGA-38-6178-01	-1.0	-1.0	-1.0
TCGA-44-2655-11	1.0	1.0	1.0
TCGA-44-2665-01	1.0	1.0	-1.0
TCGA-44-2665-11	1.0	1.0	1.0
TCGA-44-2668-01	1.0	1.0	-1.0
TCGA-44-2668-11	1.0	1.0	1.0
TCGA-44-3918-01	1.0	1.0	-1.0
TCGA-44-6145-01	1.0	1.0	-1.0
TCGA-44-6148-11	-1.0	-1.0	1.0
TCGA-44-6776-11	-1.0	-1.0	1.0
TCGA-49-6744-11	-1.0	-1.0	1.0
TCGA-50-5931-11	-1.0	-1.0	1.0
TCGA-50-5933-01	-1.0	-1.0	-1.0
TCGA-67-6217-01	1.0	1.0	-1.0
TCGA-73-4676-01	1.0	1.0	-1.0
TCGA-75-5122-01	1.0	1.0	-1.0
TCGA-91-6835-11	1.0	-1.0	1.0
LUSC			
TCGA-18-3417-01	1.0	1.0	-1.0
TCGA-22-4593-11	1.0	-1.0	1.0
HNSC			
TCGA-CV-5444-01	1.0	1.0	-1.0
TCGA-CV-6936-11	1.0	1.0	1.0

Table 3. Prediction result of the pipeline using the method of

REFERENCES

Apache Flink (2015a). FlinkML - Alternating Least Squares. https://ci.apache.org/projects/flink/flink-docs-master/libs/ml/als.html. [Online; accessed 01-October-2015].

Apache Flink (2015b). FlinkML - Connected Components. https://ci.apache.org/projects/flink/flink-docs-master/apis/examples.html#connected-components. [Online; accessed 01-October-2015].

Apache Flink (2015c). FlinkML - SVM using CoCoA. https://ci.apache.org/projects/flink/flink-docs-master/libs/ml/svm.html. [Online; accessed 01-October-2015].

Thusberg, J., Olatubosun, A., and Vihinen, M. (2011). Performance of mutation pathogenicity prediction methods on missense variants. *Human Mutation*, **32**(4), 358–368.

	Threshold 0.8	No filter
t_p	7	5
t_n	2	2
f_p	9	9
f_n	4	6
Accuracy	40.91%	31.82%
Precision	43.75%	35.71%
Specificity	18.18%	18.18%
Sensitivity	63.64%	45.45%
NPV	33.33%	25.00%

Table 4. Calculated statical values of both prediction methods (with and without filter step).