Bachelor's Thesis

Automated Discovery of Clinico-Genomic Associations: Integraded Clinico-Genomic Knowledge Discovery

Ioannis Pavlos Panteliadis

Advisor: Prof. Dimitris Plexousakis

Co-advisor: Dr. George Potamias

December, 2016

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.
I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date

Signature

Abstract

We have implemented an open source software, named *Clinico-Genomic Modeling* (CGM) which is hosted on a web server for public access using Apache Tomcat 8.0 as a container.

Multi-gene lists and single sample predictor models have been currently used to reduce the multidimensional complexity of breast cancers, and to identify intrinsic subtypes. The perceived inability of some models to deal with the challenges of processing high-dimensional data, however, limits the accurate characterisation of these subtypes

In this thesis, we are clustering the gene expressions data of 46576 different breast cancer related RNA molecules(probes) using the k-means[1] algorithm in order to *screen* and *select* the most informative probes (genes) and/or generate novel features represented as the clustered probes/genes. Based on prior published data, these RNA molecules where assigned a value between 1 and 3 based on their gene expression value. We then apply filters to the produced clusters in an attempt to discover which of the produced clusters have a high percentage of a specific value.

We then, apply Machine Learning Algorithms, provided by the Weka software, J48 Decision Tree, which is trained by the filtered features and Clinical data, in order to produce a result on a gene level regarding the patients survivability given the genome analysis.

Contents

List of Figures

1 Acknowledgments

First and foremost, I would like to thank...

- *My thesis advisor*, **Prof. Dimitris Plexousakis** (from Dept. of Computer Science, University of Crete) and *co-advisor* **Dr. George Potamias** (from Institute of Computer Science, FORTH) for providing me with the opportunity to implement this exciting project. I will always be grateful for his support and mentorship as well his endless patience with me. But above all, for introducing me to the intriguing world of Machine Learning.

  In addition, I would like to thank other members of the lab, *Alexandros Kanterakis* and *Jenny Kartsaki* for their friendship and guidance during this thesis implementation.

- *My family* for being an endless source of love and support. I am extremely blessed to have them in my life and proud to be one of them. Especially, *my grandfather, my father, my uncle, my aunt and my godfather* whom I always keep in my thoughts - *you are missed*

- *All my friends.* This long journey would not have been the same without the support of my friends, old and new. Special thanks go to: *Panos Chrisospathis, Stella Mazaraki, Katerina Chrisospathi, Tasos Mpalokas, Kostis Kleftogiorgos, Mara Mandelia, Stivi Cangonj, Rafail Troulakis, Panos Peristerakis, Blasis Choutas, Antonis Prevezanos*

2 Introduction

Breast cancer is not just one uniform disease; the specific genes and proteins that are present in the tumor cells (the molecular signature) show significant variation from person to person. These differences affect the way in which individual breast tumors respond to chemotherapy and other cancer treatments. Breast cancers can already be grouped into several different classes based on molecular signature. Physicians depend on this information when deciding how to treat each new breast cancer patient. For example:

Tumours will only respond to hormone treatment if they contain the estrogen receptor or progesterone receptor proteins that bind to female hormones and pass their messages on to the cell. Only tumours that contain the human epidermal growth factor receptor 2 (HER2) protein will respond to chemotherapy using herceptin. Directing treatments only to those patients with the best chance of responding to them avoids putting people through unnecessary and debilitating treatments, and ensures that the resources available for breast cancer treatment are used in the most efficient and beneficial way.

In our approach, we attempt to tackle the problem of finding the right treatment for a patient by automating the decision process via the available Machine Learning algorithms. In order to identify groups of genes in a gene expression associated with Breast Cancer that tend to be inherited together, we decided to employ a *One Way Clustering Algorithm* and *k-means* falls in to this category. *K-means* is an unsupervised clustering technique that partitions observations, or *features*, into k groups to increase the similarity within each cluster. *K-means* accomplishes this by using an iterative approach, first randomly choosing k observations, with the value k determined by the user, and then assigning them as the initial centroids. Centroids are the “centers” of each cluster, corresponding to the most representative observations. Next, for each feature the algorithm calculates its distance from each available centroid and assigns it to the nearest one. The centroids are then updated, typically to the average value of each cluster. The process is repeated until no further changes in clusters occur. We believe, this algorithm could reveal important and informative features that indicate the genetic architecture of breast cancer.

For this purpose we used, the widely used by the community, Weka (Waikato Environment for Knowledge Analysis) Machine Learning Workbench. An open source software issued under the GNU General Public License. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset, or in our case, called from Java code. More specifically, we used the $J48$ Decision Tree algorithm which is the implementation of the $ID3$ (Iterative Dichotomiser 3) algorithm invented by Ross Quinlan. The algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy $H(S)$ (or information gain $IG(S)$) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split by the selected attribute (e.g. $age \leq 20$, $20 \leq age < 50$, $age \geq 50$) to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before.

Given two datasets from the METABRIC[2] study, one containing clinical data on 1345 patients the other containing their respective gene expressions for 49576 different genes, we attempt to build a Clinico-Genomic Model (CGM) that combines the selected most informative clustered probes/genes and/or induced novel features,i.e., clusters of probes/genes, in order to predict the prognostic outcome for each sample/patient.

3 Background

The completion of the human genome drives us to the post-genomics era. In this environment the newly raised scientific and technological challenges push for trans-disciplinary team science and translational research. The vision is to compact major diseases on an individualized diagnostic, prognostic and treatment manner. In this context the linkage between the knowledge gained from genomics and (every- day) clinical practice (and theory as well) raises as a major challenge. The underlying Integrated Clinico-Genomic Knowledge Discovery (ICGKD) scenario and its realisation are presented in [**?**]

The microarray data are represented in a matrix with rows representing genes, columns representing samples (e.g. various tissues, developmental stages and treatments) and each cell containing a number characterizing the expression level of the particular gene in the particular sample, i.e, the gene expression matrix.

Clustering, association and classification results may reveal correlations between expression of certain genes and guide to the identification of the genetic profile of different subgroups of disease types and possibly identify new subgroups or merge together subgroups that were previously believed to be genetically separate. The identified correlations may be explored for their correspondence to corresponding patients' Clinico-histopathological profiles so that a more careful and 'revealing' examination of a disease (its genesis and development) is achieved. We believe and state that the fundamental quest of a combined clinico-genomic (transcriptional) study is to: **'explore and uncover (potential) causal relations between genomic/ transcriptional profiles and mechanisms with respective Clinico-histopathological states of diseases'**.

In the course of decision-making and disease classification, prediction models may come solely form the 'clinics' (CHPP) or, solely from the 'genomics' (GEPPs) worlds. Because CHPPs are determinable by GEPPs the quest is to identify links, relations and 'causations' between CHPPs and GEPPs in a way that refined Clinico-genomic 'individualised' disease models are identified. It is a knowledge-discovery task forwarded towards the discovery of Clinico-genomic disease theories realised by abductive and inductive inference ‘rules’. This could be done with reference to either:

- Solely Clinico-histopathological patient profiles - **CHPPs**, i.e., the clinical characterization/classification of a disease or
- Solely genomic/ transcription (i.e., gene-expression) patient profiles - **GEPPs**.

If all the above present the decision-making track in the course of a clinico-genomic research trial, the real (and most interesting) task is the **knowledge discovery** track which works in a more-or-less inverse way. That is, starting from observable Clinico-histopathological disease states, descriptions and disease- classes, the quest is to find respective genomic profiles being able to discriminate between the different disease states.

To sum up, we are combining this information and the *central dogma* of Molecular Biology which dictates that Clinico-histopathological patient profiles can be determinable by respective Gene expression patient profiles in order to answer the following question:

“Which Clinico-histopathology phenotypes relate and how with which gene-expression phenotypes?”

4 Approach

Nowadays, patients diagnosed with breast cancer have a plethora of options presented to them from their cancer care team. But the question remains, which is the best of those treatments? In this approach, we use the knowledge gained from previous cancer patients, as the train step to our J48 Decision Tree in order to generate the most reasonable decision for the patient.

In this approach, we used the publicly available files from the METABRIC study. We primarily focused on the Clinical and GeneExpression data. By analyzing these files, we were able to get a concrete idea about the gene expressions and the clinical profiles for 1356 patients.

4.1 Input to Clinico Genomic Model

The application of the ICGKD process is depended on the availability of appropriate information and data sources to store and retrieve patients' (sample-tissue) clinico-histopathology and gene-expression data.

- **Tissue /samples** extracted from specific patients (e.g. Breast-cancer) using standarised tissue-sample collection and preservation protocols - the specifics of the followed protocols are to be recorded and retrieved by the respective information system that stores and manages the respective microarray experiments to be performed and the corresponding gene-expression profiling. The collected tissue-samples are assigned (by the care team) to various **clinico-histopathological categories** that present and correspond to specific CHPPs, e.g., profiles or, classes referring to specific tumour types, stages, drug response statuses etc.

- **Gene-Expression data** Given a microarray of experimentation procedures the molecular gene expression profiles of the tissue-samples are extracted. By measuring transcription levels of genes (i.e., by microarray experiments) in an organism under various conditions we build up the respective patients' **GEPPs**, which characterize the dynamic functioning of each gene in the genome. The identification of *patterns* in complex gene expression datasets provides two benefits: (a) generation of insight into gene transcription conditions; and (b) characterization of multiple gene expression profiles in complex biological processes, e.g. pathological states.

- **Clinical and Gene-Expression Data Mediation.** Based on the availability of clinical and respective gene-expression patients' profiles – stored in respective clinical and gene-expression information systems/databases, we need mechanisms, operations and services to mediate, and seamlessly access, retrieve and uniformly model the respective heterogeneous information and data sources.

4.2 Clustering

Initially, we used the kmeans[1] clustering algorithm in order to identify characteristic clusters of genes. Clustering reduces the dimensionality of the search space, i.e., from 1000ts of genes to 10ths of gene clusters. The quest here is to organize the gene expressions into $features$ with similar profiles but also, to identify those gene clusters that shows a 'strong' and 'indicative' profile with respect to the gene-expression profiles of input samples.

For example, a cluster of genes in which all or a *high-percentage* specified by the user of the included genes exhibit a binary profile of **ON-OFF** for an adequate number of samples. Such clustered genes are indicative and suggest a potential correlation of gene-expression profiles with specific samples.

4.3 Filtering and Feature Generation

After the gene expressions have been clustered in to k different clusters, the user is prompted to supply a certain percentage to denote the percentages for the *samples coverage* and the *per sample coverage*. In the case of our experiment, we assume the patients are the samples. Therefore, by setting a filter for the *samples coverage* we are able to distinguish which clusters are **strong** enough to be considered for *feature creation*. Statistically, stronger clusters have a higher probability to contain an answer to the question we address in this thesis (see background). Namely, which drug best fits a patient's breast cancer type.

Then, we scan the generated clusters for the *per sample coverage* and if the sample is expressed in the same manner as the entire cluster (low, moderate, high) then that sample is assigned that coverage value. On the other hand, if a sample's coverage value is not equal to the cluster's value, then the sample is assigned a value of '?'. The reasoning behind this approach is due to the fact that $J48$ Decision Tree, handles '?' as a missing-value and replaces it with the mean of the other values of that attribute therefore, the generated feature will demonstrate a uniformity across the generated *features.* Finally, based on the filtering results we select the clusters that demonstrate highly confident associations between *GEPPs* and *CHPPs.*

4.4 Generating the Decision Tree

The question then is: **"which metagenes/indicative GEPPs relate and how to which CHPPs?"**. This may be accomplished with the aid of an **ASSOCIATION RULES MINING (ARM)** methodology in order to automatically discover 'highly confident' *associations* between GEPPs and CHPPs.

We decided to use the Weka[3] workbench to generate the $J48$ Decision Tree. The decision to use this predictive machine learning model due to it's ability to decide the target value of a new sample based on various attribute values from the original data. The internal nodes of a decision tree denote the different attributes, the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. The $J48$ Decision Tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained. For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess.

5 Experiments

In this thesis we used a MacBook Pro running Mac OS Sierra on a 22 nm "Ivy Bridge" 2.9 GHz Intel "Core i7" processor (3520M), with two independent processor "cores" on a single silicon chip, a 4 MB shared level 3 cache, 16 GB of 1600 MHz DDR3L SDRAM (PC3-12800) installed in pairs (two 8 GB modules) and a 500 GB RAID0 Partitioned installed in pairs hard drive (two 250GB modules) Solid State Drives (SSD) . In order to reproduce our experiment you can view the project source code from the project's *GitHub* repository https://github.com/ppanteliadis/ClinicoGenomicModelling or download the *.war* from Dropbox https://www.dropbox.com/sh/ye06u6yckvin0ks/AAD31vaOtKoOhm2-bhnJ2dGna?dl=0.

To run our project open your version of Eclipse (download latest version from here: https://www.eclipse.org/downloads/) :

1. Go To File > Import > Type in the input source WAR and select it. The project will load, but most likely with errors.

   The errors are caused due to the fact that most machines don't have the

   If the project has no errors, go to step 2, if not, go to step 3.

2. a) Right-click the project folder, click Properties

   b) Choose Java Build Path

   c) Click Add External JARs. . .

   d) Browse in the Thesis folder for **javax.servlet-api.jar** and select it

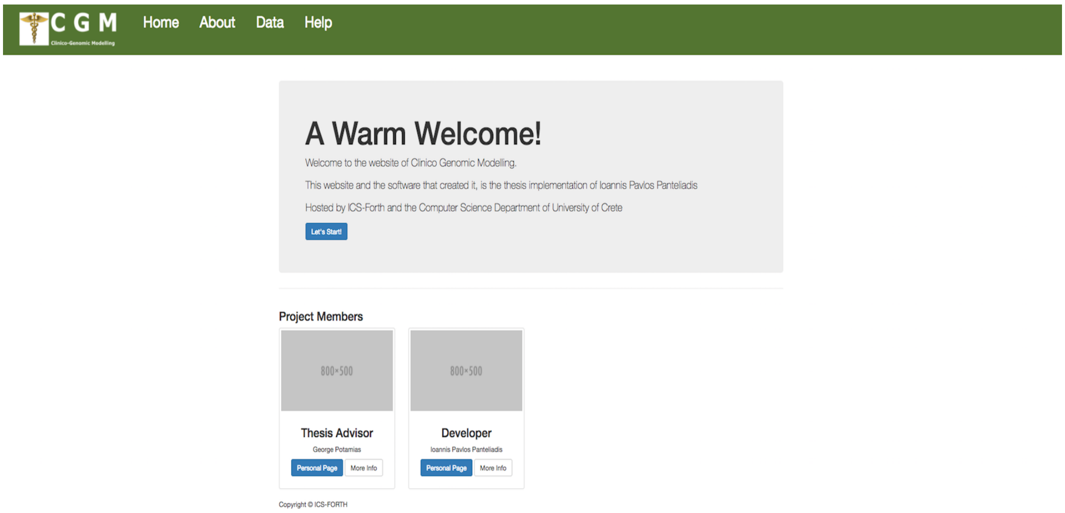
   e) Click OK to update Build Path.

3. Right-click the project folder

   a) Click Run As > Run on server

   b) Click Finish

In this section we provide screenshots of a test run for 1345 patients and 49576 Gene Expression data as well as 16 clinical data for the same patients. Our goal is to demonstrate the simplicity and robustness our software.



A Warm Welcome!

Welcome to the website of Clinico Genomic Modelling.

This website and the software that created it, is the thesis implementation of Ioannis Pavlos Panteliadis

Hosted by ICS-Forth and the Computer Science Department of University of Crete

Project Members

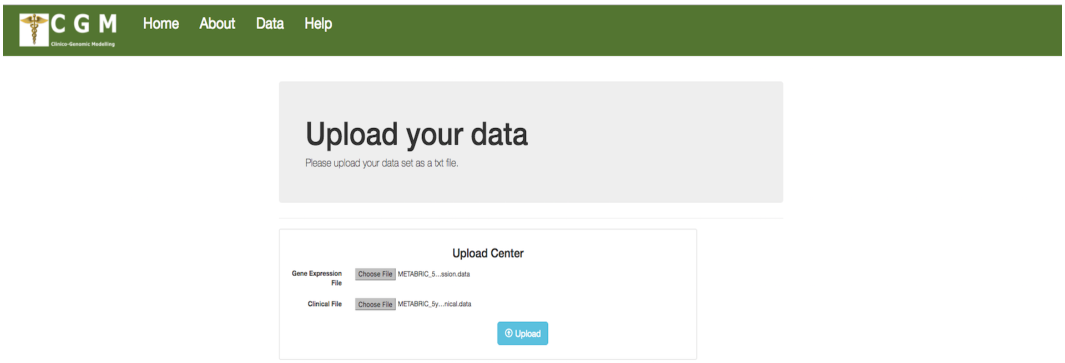
Thesis Advisor
George Potamias

Developer
Ioannis Pavlos Panteliadis

Copyright © ICS-FORTH

Figure 1: Clinico Genomic Modelling welcoming screen.

In this screen the user is simply introduced to the model. All he has to do is click the $Let'sStart$ button to initialize the process. By returning to this screen, the model will be re-initialized, so all progress or tests made, will be lost.



Upload your data

Please upload your data set as a txt file.

Upload Center

Gene Expression File: METABRIC_5...ssion.data

Clinical File: METABRIC_5y...nical.data

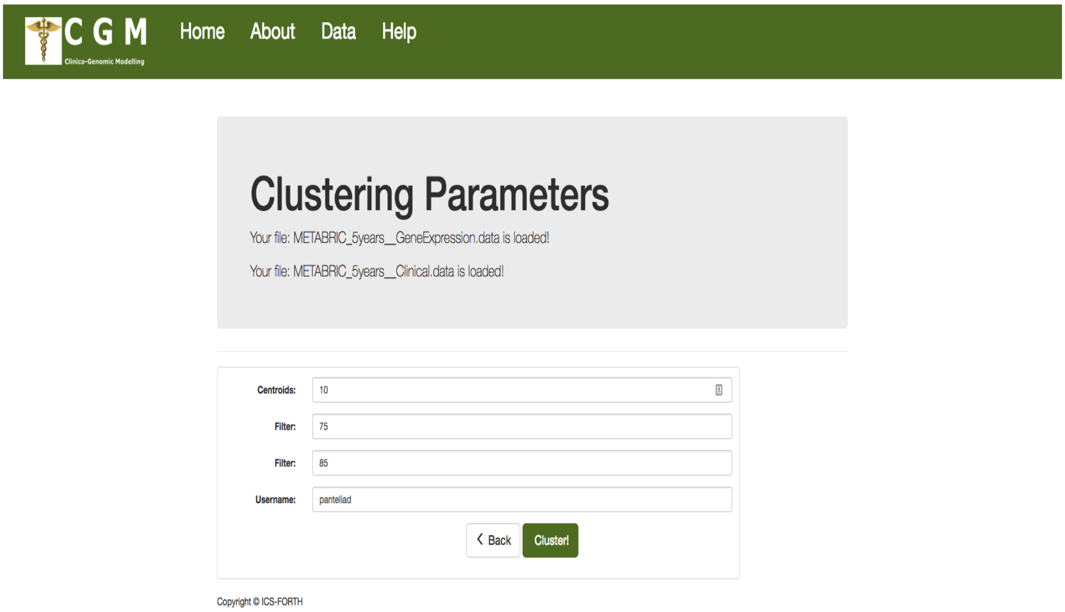
Figure 2: Uploading user data.

Again, a very straight forward window in which the user will have to supply the model with the data he will be using. Currently, the model needs **two** very specific files in order to compute. As it has previously been stated in this thesis, a file with clinical and gene expression data is needed in order for the model to operate.

In our case, we supplied the model with the publicly available

- **METABRIC_5years__GeneExpression.data** and
- **METABRIC_5years__Clinical.data**

files.



Clustering Parameters

Your file: METABRIC_5years__GeneExpression.data is loaded!

Your file: METABRIC_5years__Clinical.data is loaded!

Centroids: 10

Filter: 75

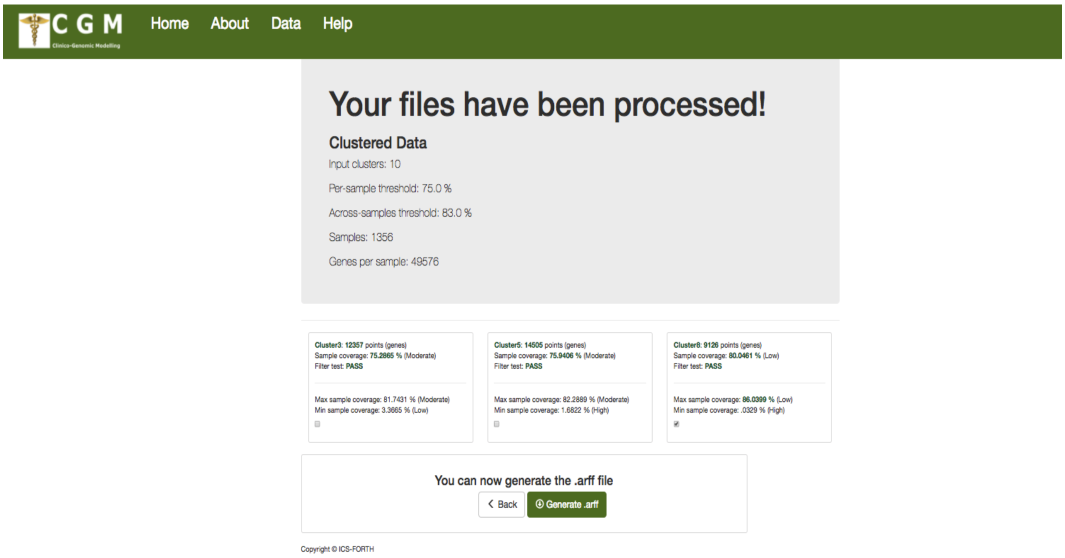
Filter: 85

Username: panteliad

Copyright © ICS-FORTH

Figure 3: Clustering parameters.

After the successfull uploading of the files on the server, the user is prompted to enter the clustering parameters which he is interested in. **Note:** A test with a high filter may **not** produce strong clusters. As a precausion, we used a *relatively* low filter of *75%* in this demonstration in order to make sure that at least *one* of the clusters will be strong enough in order to be selected for designing the *DecisionTree* as it was mentioned in *Chapter 3.4* of the thesis.



Your files have been processed!

Clustered Data

Input clusters: 10

Per-sample threshold: 75.0 %

Across-samples threshold: 83.0 %

Samples: 1356

Genes per sample: 49576

Cluster3: 12357 points (genes)
Sample coverage: 75.2865 % (Moderate)
Filter test: PASS

Max sample coverage: 81.7431 % (Moderate)
Min sample coverage: 3.3665 % (Low)
☐

Cluster5: 14505 points (genes)
Sample coverage: 75.9406 % (Moderate)
Filter test: PASS

Max sample coverage: 82.2889 % (Moderate)
Min sample coverage: 1.6822 % (High)
☐

Cluster8: 9126 points (genes)
Sample coverage: 80.0461 % (Low)
Filter test: PASS

Max sample coverage: 86.0399 % (Low)
Min sample coverage: .0329 % (High)
☑

You can now generate the .arff file

Copyright © ICS-FORTH

Figure 4: Generating WEKA file.

After a short delay, the user will be notified that the Clustering process has been completed. At this point, the user will be able to view some *descriptive statistics* about the files he just uploaded on the main frame of the website.

By scrolling further down, all the clusters that were created will be labeled as Cluster (*0 to k-1*) the number of points they include inside them, the sample coverage and their percent value, the result for the per sample filter, the maximum and minimum sample coverage percentages and the value these values represent. Note here, that only the cluster whom we characterize as **strong** will be displayed to the user. After the creation, the user can then select, which cluster features to be created for him to further analyze them in the *decision making process*.

We then used the generated feature file and run a 10-fold validation using the Weka tool[3] for the METABRIC (49576 gene expression data * 1365 patients) data, used in this experiment. The results of the stratified cross validation are demonstrated below:

Table 1: Stratified Cross Validation

Correctly Classified Instances	995	73.3776 %
Incorrectly Classified Instances	361	26.6224 %
Kappa statistic	0.2395	
Mean absolute error	0.3431	
Root mean squared error	0.4429	
Relative absolute error	85.4313%	
Root relative squared error	98.8501%	
Total Number of Instances	1356	

Table 2: Detailed Accuracy By Class

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.313	0.104	0.536	0.313	0.395	0.254	0.667	0.447	bad
0.896	0.687	0.772	0.896	0.829	0.254	0.667	0.799	good

Table 3: Weighted average

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
0.734	0.525	0.706	0.734	0.709	0.254	0.667	0.701

Table 4: Confusion Matrix

a	b	classified as
118	259	a = bad
102	877	b = good

6 Conclusion

We used the Weka tool [3] to validate our approach. By running the generated *Attribute-Relation* file through a **10-fold cross validation**.

Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on *new* data. This is the basic idea for a whole class of model evaluation methods called cross validation.

The **holdout method** is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The function approximator fits a function using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

V-fold cross validation is one way to improve over the holdout method. The data set is divided into v subsets, and the holdout method is repeated v times. Each time, one of the v subsets is used as the test set and the other V-1 subsets are put together to form a training set. Then the average error across all v trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set V-1 times. The variance of the resulting estimate is reduced as v is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch V times, which means it takes V times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set V different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

Clinico-Genomic Modelling (CGM) can be used to estimate a prediction of whether a patient is qualified for a certain Breast Cancer treatment. Due to its handling of large information in linear time $O(n*k*i*d)$, $n = points$, $k = clusters$, $i = iterations$, $d = attributes$. plus an additional $O(h)$ for the construction of the decision tree. These times makes our software particularly suited to be used by specialists in the field of medicine, in order to provide their patients with more accurate prediction regarding the most optimal prediction about the treatment course for their cancer.

Bibliography

[1] T. Kanungo, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 24, July 2002.

[2] I. Rezaeian, E. J. Mucaki, K. Baranova, H. Q. Pham, D. Angelov, A. Ngom, and L. Rueda, "Predicting outcomes of hormone and chemotherapy in the molecular taxonomy of breast cancer international consortium (metabric) study by biochemically-inspired machine learning.," *F1000Research*, August 2016.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutermann, and I. H. Witten, "The weka data mining software: An update," *ACM SIGKDD Explorations Newsletter*, pp. 10–18, June 2009.

Bibliography

[1] T. Kanungo, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 24, July 2002.

[2] I. Rezaeian, E. J. Mucaki, K. Baranova, H. Q. Pham, D. Angelov, A. Ngom, and L. Rueda, "Predicting outcomes of hormone and chemotherapy in the molecular taxonomy of breast cancer international consortium (metabric) study by biochemically-inspired machine learning.," *F1000Research*, August 2016.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutermann, and I. H. Witten, "The weka data mining software: An update," *ACM SIGKDD Explorations Newsletter*, pp. 10–18, June 2009.