ICT-2011-288048

EURECA

Enabling information re-Use by linking clinical Research and CAre

IP
Contract Nr: 288048

Deliverable: D5.2 State-of-the-art review of existing methods and tools for hypothesis generation and association studies

Due date of deliverable: (01-03-2013)
Actual submission date: (09-04-2013)

Start date of Project: 01 February 2012 Duration: 42 months

Responsible WP: UOXF

Revision: <outline, **draft,** proposed, accepted>

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Ruud van Stiphout	UOXF	ruud.vanstiphout@oncology.ox.ac.uk
Francesca Buffa	UOXF	francesca.buffa@imm.ox.ac.uk
Stefan Rueping	IAIS	stefan.rueping@iais.fraunhofer.de
Andre Dekker	MAASTRO	andre.dekker@maastro.nl
Scott Marshall	MAASTRO	m.scott.marshall@maastro.nl
Lefteris Koumakis	FORTH	koumakis@ics.forth.gr

0.2 Documents history

Document version #	Date	Change
V0.1	06.02.2013	Starting version, template
V0.2		Definition of ToC
V0.3	28.03.2013	First complete draft
V0.4	28.03.2013	Integrated version (send to WP members)
V0.5	04.04.2013	Updated version (send PCP)
V0.6	04.04.2013	Updated version (send to project internal reviewers)
Sign off		Signed off version (for approval to PMT members)
V1.0		Approved Version to be submitted to EU

0.3 Document data

Keywords	Methodological review, association, hypothesis generation
Editor Address data	Name: Ruud van Stiphout Partner: UOXF Address: Weatherall Institute of Molecular Medicine, University of Oxford Phone: +44 (0)1865 222440 E-mail: ruud.vanstiphout@oncology.ox.ac.uk
Delivery date	

0.4 Distribution list

Date	Issue	E-mailer

Table of Contents

1 INTRODUCTION

Data mining plays a crucial role in the overall goal of EURECA. The aimed link between clinical research and clinical care systems requires methods to extract the relevant data and patterns out of the overwhelming large amounts of available data. Furthermore, this information needs to be presented to the health care professional and the patient in a highly interpretable fashion. Data mining in EURECA is mainly developed and applied in scenarios such as diagnostic classifiers, outcome prediction and hypothesis generation but other scenario's will use them also, like for example the text mining from EHRs for extraction of meaningful information to populate research databases. This deliverable will describe the state-of-the-art methods involved in data mining with the specific focus on EURECA specific aims and tools.

The structure of this deliverable is as follows. Section 2 summarizes which technical scenarios use data mining techniques and which type of data mining. Section 3 then provides a review of the useful methods, including references to available tools. Section 4 describes specifically for each of the data mining involved technical scenarios which methods are used or suggestions will be made based on similar published tools. Section 5 provides an overview of the relevant tools and platforms to develop the data mining methods in. In general, many references to tools or packages are provided with the focus in the free software package R. This choice is motivated in section 5.

2 DATA MINING IN EURECA

This section provides an overview of all the data mining methods used in EURECA in Table form (Table 1).

Table 1. Overview of technical scenarios and their corresponding classification in data mining techniques

General scenarios	Partner scenarios	Technical scenario	Responsible Partners	Data pre-processing	Similarity learning	Association rule discovery	Classification	Clustering	Text mining	Change detection	Time dependent data & Data stream mining	Privacy preserving DM	Distributed data learning
Information	SIT2 VUA2 UdS3	Personal medical information recommender	StoneRoos	x		x			x		x		
	FORTH1	Export from an HER to a PHR	FORTH										
	UdS2	Data mining for consultation	FhG IAIS			x			x		x	x	
	VUA1 C-P1	Contextualized overview	VUA						x				
Investigation *Guidelines*	UdS4 Maastro1	Update of guidelines	VUA						x		x		
	UOXF1 Maastro4	Train, validate and updating a diagnostic classifier	UOXF	x		x	x	x	x	x	x	x	x
Investigation *Protocol & research*	UdS5	Broad consent	Custodix										
	UdS6	Hypothesis generation	UOXF	x	x	x	x	x		x	x		
	VUA3 UdS7	Protocol feasibility	Philips										
Selection & Recruitment *Treatment choice*	UdS8	Microbiology SAE	FhG IBMT										
	Maastro2 Maastro3	Outcome prediction	UOXF	x		x	x	x	X	x	x	x	x

	UOXF1 Maastro4	Use a diagnostic classifier	UOXF	x									
Selection & Recruitment *Patient trial recruitment*	Maastro5 Maastro6 IJB1 BIG1 BIG2 UOXF2 UdS9	Trial recruitment	Custodix	x	x				x	x		x	x
Reporting	IJB2	Reporting episodes of febrile neutropenia	IJB										
	IJB3 IJB4 IJB5	Cancer registry and tumour bank reporting	IJB										
	Maastro7	Pre-filling of CRF and AE reports	UPM										
	Maastro8 UdS10 UdS11	Automatic detection and reporting of SAEs/SUSARs	FhG IBMT	x		x	x		x	x	x		
Long-term follow-up	IJB6 IJB7 UOXF3 IJB8 IBMT1 UdS12*	Long-term follow-up and patient diary*	FORTH						x		x	x	
Economic analysis	UdS13	Analyse economic data between different procedures	FhG IAIS	x		x	x			x		x	x

3 DATA MINING APPROACHES AND METHODS

The previous sections described the different technical scenarios and use cases that use data mining techniques to accomplish their aim. Below the state-of-the-art data mining techniques are described in more detail including references to use cases and literature. This is an extension on the brief data mining overview which was given in D5.1. The described methods are not all methods available, but only the main areas that will be useful and used in EURECA. We will refer to this proposed classification throughout the document, starting with Table 1. This table provides only a quick reference overview. Methods classified per scenario often overlap, and some of these are more approaches then methods. But the focus, and particularly the EURECA application, is different, as described in this section. We can characterise four classes of data-mining approaches:

- Supervised: the data is labelled (e.g. outcome). Very important in almost all DM techniques, but most known for classification.
- Unsupervised: unlabelled data. Typically unsupervised clustering algorithms can be used to find groups of patients with similar characteristics.
- Mixed: more complex techniques which use semi-supervised learning.
- Knowledge based: these techniques use knowledge and reasoning in their tasks. This knowledge is often represented by rules, frames or scripts.

3.1 Data pre-processing

New clinical trials include techniques such as high-throughput assays and imaging techniques which produce a very large amount of data points/variables. Thus, data pre-processing has become a very important step in data analysis. The main pre-processing methods applied to medical data are:

- **Outlier detection:** out-of-range values entered wrongly in the dataset or by measurement error can affect DM algorithms. Detecting combinations of data which are unlikely or impossible is also important, e.g. patient gender is male for cervical cancer patients.
- **Missing values:** values not present in the dataset can be dealt with according to the task. Imputation (substituting) of the values using specific algorithms is common for classification problems, but not always necessary since some classification methods can deal with missing values.
- **Normalization:** since the range of values of raw data varies widely, in some DM algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.
- **Dimensionality reduction:** Reasons to reduce the number of variables in a dataset used for data mining are: 1) Many of the variables in the available datasets are correlated and not independent. To avoid over fitting and improve model performance, i.e. prediction performance in the case of supervised classification and better cluster detection in the case of clustering 2) In high dimensional datasets the false discovery rate can be high due to multiple testing, meaning that there is a risk that one will find a significant predictor b

chance 3) to provide faster and more cost-effective models 4) to gain a deeper insight into the underlying processes that generated the data. Dimensionality reduction approaches can be applied before the analysis or, for example in classification problems, whilst building the classifier. This helps gaining statistical power in analyses where usually the number of variables is much higher than the number of cases.

This section will now focus on dimensionality reduction using feature selection since in EURECA we deal with high dimensional datasets and due to the complexity of this domain, feature selection is one of the most common pre-processing steps for extracting knowledge from genomic data.

3.1.1 Feature selection

Reduction of the *dimensionality* of data is a well-known problem in machine learning and data mining, denoted as *feature selection*.[1] In its general form the problem could be stated as follows:

> *Given a set of features (attributes or descriptors, i.e., molecular markers) m and a target variable T (i.e., phenotypic classes); Find an optimal subset r of features, $r \subset m$ that achieves maximum classification performance over T for a given set of predictors (classifiers) and respective classification performance metrics (e.g., predictive accuracy, sensitivity, specificity etc).*

A strategy for feature selection should implement a *search* through the space of possible feature subsets that addresses the following landmark questions[2]:

- *Where to start and to which direction the search*?
  - Begin with an empty set and start adding individual (or subset of) 'useful' features or, begin with all (or part of) the features and start removing 'useless' features
- *How to assess the usefulness of features*?
  - The two main strategies are the *filter* and *wrapper* approaches (see below)
- *How to search*?
  - As an exhaustive search is intractable (especially for huge dimensional domains, like microarray gene expression data) *heuristic* search methods should apply
- *When to stop the search*?
  - Adding of removing features could stop when none of the alternatives improves performance (e.g., predictive accuracy).

In the light of these observations, a general feature selection process could be realized in three basic steps (**Error! Reference source not found.**): generation, evaluation, stopping criterion, and validation (on external test cases) of the selected feature subset[3].

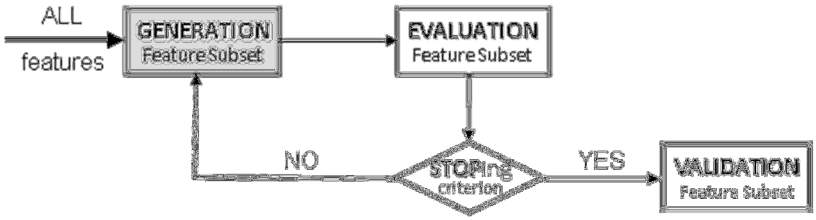

Figure 1. The generalized feature selection methodology: components and operational flow – the "generalization" component is shaded to point its major importance in the underlying feature space search process.

Given a domain with *m* input features, finding the best feature subset in an exhaustive-search mode (i.e., in a 2^m space of feature subsets) is known to be NP-hard[4], with the search to become quickly computationally intractable.

3.1.1.1 Basic Feature Selection Strategies (Filters and wrappers)

The intractability of a complete features' space search, forces us to concentrate on *heuristic search* approaches where, with the risk of losing solutions, optimality could be approximated. The long-term machine learning and data mining research have elaborated on two major families of feature selection methodologies realized by different heuristic space search strategies: the *filter* and the *wrapper* methods.

- ***Filters***: In the filter-based feature selection (FFS) approach the feature space is not explored via the use of an induction algorithm; instead the features (or, subset of features) are evaluated and selected on the basis of their statistical properties. In most cases the evaluated property relates to the power of the features to discriminate between the classes, as assessed by respective feature scores or *ranks*. FFS is utilized as a pre-processing step in order to select the most characteristics and/or discriminant (with respect to the available classes) features. Then, a particular classifier may be applied on the reduced dataset. So, FFS techniques are not classifier specific. In filter-based approaches, selection implies deployment of a scoring or, *ranking* procedure, to measure the power of a gene to discriminate between the different sample categories.

- ***Wrappers****:* In the wrapper-based feature selection (WFS) approach the feature subset selection algorithm exists as a wrapper around the utilized induction algorithm, that is: the induction algorithm itself (considered as a 'black-box') is used as part of the feature subset evaluation function. In other words, the feature selection component is embedded in the algorithm, carrying of course the classifier's bias. In the worst, exhaustive search case - where all different feature subsets should be evaluated, WFS approaches exhibits an exponential to the number of features *m* (time) complexity, $O(2^m)$. As already noted this is impractical for gene expression studies. More economical wrapper-based features selection algorithms have been proposed. One such simple algorithm is implemented by *backward elimination* where the search starts with the full set of features and proceeds by greedily removing features until performance starts to degrade. Another option is *forward selection* where, one starts with the empty set and proceeds by greedily adding features until no further improvement can be achieved. Due to their high computational complexity WFS techniques have not received much interest. In order to cope with this, a reduction of the feature space is applied first (e.g., following a filter approach), followed by the wrapper or embedded component on the reduced data set, hence fitting the computation time to the available resources..

Combined *hybrid* filter-wrapper techniques have been also proposed. Such techniques base their selection strategy on a pre-ordered ranking of features followed by an incremental feature selection process. With this approach, the respective computational burden is relaxed at great extent.

3.1.1.2 Feature selection for EURECA

The advent of genomic and proteomic high-throughput technologies enabled a 'systems level analysis' by offering the ability to measure the expression status of thousands of genes in parallel, even if the heterogeneity of the produced data sources make interpretation especially challenging. The high volume of data being produced by the numerous studies worldwide, post the need for a long-term initiative on bio-data analysis in the context of 'translational bioinformatics' research.
In the context of EURECA feature selection is relevant for applying data mining on clinical data but essential for the identification of relevant biomarkers that accurately predict risks in patients and to validate new hypotheses using large (genomic) studies. For such a high dimensional domain, where one must explore the space of 2^{30000} gene subsets, an exhaustive search is practically impossible. It is proved that, in the case that the evaluation criterion possesses the *monotonicity* property (i.e., a subset of features should be not better than any larger set that contains the subset) an optimal subset of features could be found without evaluating the whole space of 2^m feature subsets[5].

3.2 Similarity Learning (SL)

Similarity Learning consists of classification on pairwise similarities. In contrast to other machine learning methods, similarity learning does not assume that objects are well represented in a Euclidean feature space. This is useful for problems in bioinformatics, information retrieval and many other areas with diverse object representations. In EURECA for example we want to find similar clinical trials. Since clinical trials more frequently now include pathology, genomic and imaging data, the representation of semantic similarity will need to be defined in extremely complex data space.

A typical application of similarity learning is a recommender system. They attempt to recommend information items that are likely to be of interest to the user. "Typically, a recommender system compares a user profile to some reference characteristics, and seeks to predict the 'rating' or 'preference' that a user would give to an item they had not yet considered."[6] Many algorithms used for recommender systems and gene pattern recognition are based on distance measures. The distance indicates the similarity of information items. Then, those items are recommended that are "closest" to match the user profile. User interaction can be used as feedback to improve the similarity models, in that the system will observe the choices made by the user with regard to the "similar items" offered, extending the list of similar items as input for similarity learning.

The key idea of similarity learning is to replace fixed distance functions by learning a function that produces a non-negative real number for any pair of examples. The intended semantic is that the higher this number the more similar the two examples are. The training data that the function learns from consist of example pairs labelled as similar or dissimilar.

The learning framework is generic, in that information items are considered as structured objects (of arbitrary nature). For instance, the information item may be a document with a substructure given by “title”, “author”, “abstract”, “main text”, and, eventually, “metadata”. Typically, similarity learning proceeds by attaching basic distance measures to the atomic components, eventually as well to structural properties, and then by learning weight coefficients applied to the basic distance measures. More formally, let D be a set of base distance measures. Then the distance of two “points” x, y is defined by where are the weight coefficients. Points are typically n-tuples of atomic components but may as well be structures such as trees or graphs.
Basic distance measures are distinguished by the type of the atomic component. A number of distance measures, well known from the literature[7], are listed below:

- String
  - LevensteinSimilarity, BlockDistance, DiceSimilarity, JaroWinklerSimilarity, MatchingCoefficient, JaccardSimilarity, CosineSimilarity on words, ChapmanLengthDeviation, ChapmanMatchingSoundex, ChapmanMatchingSoundexSpanish, Jaro, MongeElkan, NeedlemanWunch, OverlapCoefficient, QGramsDistance, SmithWaterman, SmithWatermanGotoh, SmithWatermanGotohWindowedAffine, SoundexEnglish, SoundexSpanish,
- Numbers, Number series
  - EuclideanDistance, CosineSimilarity, CamberraDistance, ChebychevDistance, CorrelationSimilarity, JaccardSimilarity, ManhattanDistance
- Boolean
  - Jaccard Similarity, Dice Similarity, Matching Koeffizient, Cosinus Similarity
- Text
  - Cosine distance on n-grams
- Structured Data
  - Hierarchies in the data may be reflected by taking the path length into account.
  - Hierarchy in structures such as trees or graphs may be reflected by taking the path length into account.

The result of similarity learning is a similarity model that consists of the weight coefficients for the basic distance measures.

Due to the variety of information objects, the question of how to define a similarity for them is a challenging task. It is even more complicated in case the definition of similarity depends on user needs. It makes no sense to generalize a recommendation function among multiple users. There is a requirement for an easy-to-use service that each user can create a similarity model according to his/her particular preferences.

At the same time, a typical user will be unwilling to spend a lot of time to set up a recommendation system. The process of obtaining labelled data is costly in terms of time and manual effort. In order to start learning with n examples, the user needs to give his feedback for $n * (n-1)$ object pairs. Hence, he should be only asked for input that he can give quickly and correctly. In particular, it is very favourable to ask the user only questions regarding specific instances, for which domain experts can usually give very concrete feedback. As an example, when recommending papers to read, it is

better to ask the user “is this paper relevant to you?” instead of “do you like to see more papers of the same author?” In order to reduce the user’s efforts in labelling, the selection of a small set of pairs that is informative enough to create an accurate model is necessary. Intelligent sampling strategy that selects the most ‘interesting’ pairs from a pool of unlabelled data to show them to the user exist[8].

3.3 Association Rule Discovery

Association rule discovery (ARD) considers the problem of discovering association rules between items in a large databases; it has been applied extensively for example to databases of sales transactions but less so to the clinical or medical sciences. Algorithms have been proposed and tested mainly for categorical data and less for numerical data. This is because it does not perform well for numeric data[9-14]. An ARD algorithm requires a collection of instances as input and provides rules to predict the values of any attribute(s) (not just the class attribute) from values of other attributes as output.

ARD can also be used to do an integrative analysis of microarray data. The approach can integrate gene annotations and expression data to discover intrinsic associations among both data sources based on co-occurrence patterns, which can help in determining the cause of mutation in tumours and diseases. Typical annotations are metabolic pathways, transcriptional regulators and Gene Ontology categories. Previous studies automatically extracted associations revealing significant relationships among these gene attributes and expression patterns, where many of them are clearly supported by recently reported work.[14, 15]

Available R-package: http://cran.r-project.org/web/packages/arules/index.html

3.4 Classification

Classification is a mining technique based on machine learning; it is used to classify each item in a set of data into one of predefined sets of classes or groups. The data classification process involves a learning phase and classification phase. In the learning phase a set of training data are analysed by a classification algorithm; then in the classification phase data are used to estimate the accuracy of the classification rules. If the accuracy is acceptable the rules can be applied to any new and similar data. The classifier-training algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. The algorithm then encodes these parameters into a model called a classifier. There are different algorithms to do this classification, each of them having their strengths and weaknesses, and their optimal data type suitability. A set of well-known and state-of-the-art extensions and applications of these algorithms is discussed in this subsection.

3.4.1 Support vector machines

This method was introduced in 1995 [16] and is since then widely used in bioinformatics and other fields due to its high accuracy, the ability to deal with high-dimensional data (such as gene expression) and the flexibility of modelling diverse sources of data. SVMs belong to the general category of kernel methods. A kernel method only depends on the data through dot-products. In that case, a dot product can be computed in a possibly high dimensional feature space by replacing the dot-product

with a kernel function. This has two advantages: 1) the ability to generate non-linear decision boundaries using methods designed for linear classifiers 2) The use of kernel functions allows the user to apply a classier to data that have no obvious fixed-dimensional vector space representation. In general SVMs are sensitive to the way features are scaled. Therefore it is essential to normalize the data because the accuracy of the classifier can degrade severely. As in most other classifiers, feature selection is important in SVMs; not necessarily to improve accuracy, but to understand better the data and the classification results.

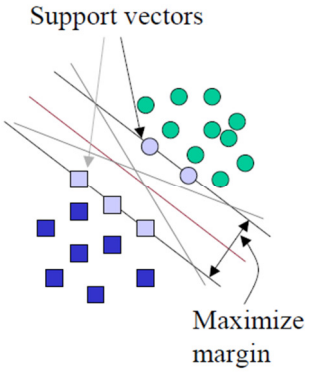

Figure 2. Concept of SVMs: maximizing the margin between two labelled groups of samples. The samples that define the margin are called support vectors. This is an extreme example with no misclassifications. In practice, soft margins are used that take misclassification rate into account in the maximizing process.

Advantages:

- Accurate and robust classification results on different data types
- Uses a subset of training points in the decision function so it is also memory efficient.
- Expert knowledge can be implemented by designing the kernel
- Convex optimisation problem (no local minima in optimisation process)
- Non-linear modelling ability
- Strong theoretical basis

Disadvantages:

- Classification in a black box fashion, i.e. they do not provide the user much information on why a particular prediction was made.
- Most SVMs are two-class classifiers, although multi-class classifiers exist but they are computationally more expensive.
- SVMs do not directly provide probability estimates

Published applications:

- MicroRNA profiling to distinguish lung cancer patients from healthy controls.[17]
- Gene and microRNA expression predicts nodal involvement in breast cancer.[18]
- Prediction of event-free-survival for neuroblastoma patients using miRNA expression.[19]

3.4.2 Random forests

3.4.2.1 Decision trees

To understand the concept of random forests the decision tree classifier needs to be explained. Decision trees try to find ways to divide the universe into successively more

subgroups (creating nodes) until each addresses only one class or until one of the classes shows a clear majority that does not justify further divisions, generating in this situation a leaf containing the class majority (example: Figure 3). The algorithm starts with a training set in which the classification label is known for each record. The algorithm then systematically tries to break up the records into two parts, examining one variable at a time and splitting the records on the basis of a dividing line in that variable. The objective is to attain an as homogeneous set of labels as possible in each partition. This splitting or partitioning is then applied to each of the new partitions. The process continues until no more useful splits can be found. In this way a decision tree is constructed which is highly interpretable. The heart of the algorithm is the rule that determines the initial split rule. In general, every possible split is tried and considered, and the best split is the one which produces the largest decrease in diversity of the classification label within each partition. Another concept which is applied in the development of decision trees is pruning. This is the process of removing leaves and branches to improve generalizability of the tree for new data because in some nodes the populations are not representative anymore.

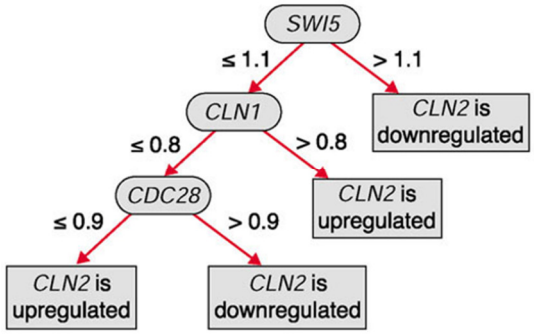

Figure 3. Simple example of a decision tree for CLN2 gene regulation by three explaining genes[20]

3.4.2.2 Supervised random forests

A random forest actually grows a collection of many classification trees. To classify a new object, this object is input for each of the trees in the forest. After each tree provides a classification, the forest picks the classification having the most votes. Each tree is grown as follows:

1. If the number of cases in the training set is *N*, sample *N* cases at random from the original data (with replacement) as a training set.
2. If there are *M* input variables, a number *m<<M* is specified such that at each node, *m* variables are selected at random out of the *M* and the best split on these *m* is used to split the node. The value of *m* is held constant during the forest growing.
3. There is no pruning, so each tree is grown to the largest extent possible.

The forest error rate depends on:

1. The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
2. The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

The task is to find the optimal range for the value of *m* because correlation and strength go up or go down simultaneously when altering *m*. To find the optimal *m*, the oob (out-of-bag) error rate is in general used. This error rate provides an internal validation of the test set error: A different bootstrap sample from the original data is used to construct each tree. One-third of this bootstrapped data is left out to test on the same tree. These classified "new" cases are compared to their actual class and the error estimate is averaged over all cases, providing an unbiased error rate.

Advantages:

- Very accurate method overall
- Can handle high dimensional inputs
- Provides variable importance estimates
- Can deal with missing data accurately

Disadvantages:

- Risk of overfitting in noisy classification tasks
- For data including categorical variables with different number of levels, random forests are biased in favour of those attributes with more levels.

Published applications:

- Finding specific mutations for melanomas using RF.[21]
- Identification of microRNAs associated with overall patient survival in neuroblastoma.[22]

3.4.3 Bayesian networks

A Bayesian network (BN) is a graphical model that encodes probabilistic relationships among variables of interest. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables (Figure 4). There are three general tasks for the development of BNs:

1. Structure learning: the structure of the network can be provided by an expert, be learned from the data, or both.
2. Parameter learning: given the structure of the network, for each node the variable distribution needs to be estimated given the information of the "parents" nodes.
3. Inferring unobserved variables: the network can be used to find out updated knowledge of the state of a subset of variables when other variables (the evidence variables) are observed.

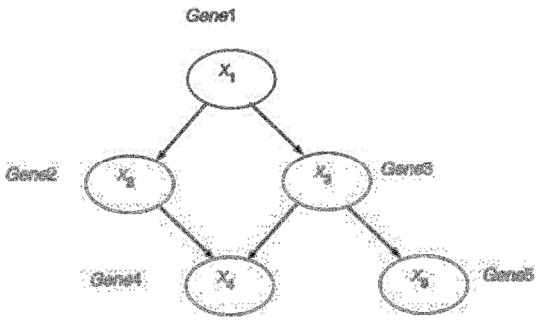

Figure 4. A Bayesian Network representing a hypothetical gene-regulation pathway. This structure of a hypothetical Bayesian Network contains five nodes. The probabilities associated with this network structure are not shown. The network structure indicates that Gene1 can regulate (influence) the expression level of Gene3, which in turn can regulate the expression level of Gene5.[23]

Advantages:

- BN can deal with incomplete datasets
- Causal relationships can be learned
- They facilitate the use of prior knowledge
- No data preprocessing required to avoid overfitting
- Easy to interpret visualization of the model

Disadvantages:

- Very sensitive to the (subjective) structure of the network
- Costly computational task
- Not all BN software can deal with continuous data (discretization required)
- Feedback effects cannot be included in the network (acyclic nature of BNs)

Published applications:

- Modeling local failure in lung cancer using clinical, dosimetric variables and blood biomarkers.[24]
- Discriminating responders and non-responders for head and neck cancer patients with specific gene clusters.[25]

3.4.4 Artificial neural networks

The concept of a neural network (NN) learning algorithm is inspired by the structure and functional aspects of biological neural networks. Computations are structured in terms of interconnected artificial neurons, which are usually non-linear in nature. These networks are constructed with an input layer with all the variables, one or more hidden layers, and an output layer which produces the estimation of your target outcomes (Figure 5).

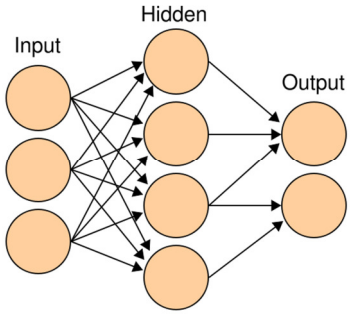

Figure 5. Example of an artificial neural network with one hidden layer (http://offthelip.org)

Advantages:

- Powerful technique utilized across scientific disciplines.
- Theoretically well suited to non-linear processes

Disadvantages:

- Not transparent and therefore hard to interpret results ("black box" concept)
- Technically difficult to understand
- Empirical nature of model development
- Computationally expensive

Published applications:

- NN was better in predicting breast cancer survival than decision trees and logistic regression using gene expression data.[26]
- Modelling interaction between mRNA and microRNA using fuzzy neural networks[27]

3.4.5 Data types

The selection of a classifier mainly depends on the input data that is provided. One can distinguish continuous data and discrete data, which can be categorical (nominal) or ranked (ordinal). Table 2 shows some examples of these data types.

Table 2. Cancer related examples of categorical and continuous data for different sources

Input origin	Discrete (category/ranked)	Continuous
Clinical	Tumour stage Health performance score (1-4)	Age Blood pressure
Imaging	Tumour invasion >20mm (yes/no) Nr of lymph nodes (0, 1-3, >3)	Tumour heterogeneity Tumour sphericity
Genomics	Copy number	Gene expression levels RNA expression levels
Treatment	Chemo administration (yes/no)	Radiotherapy dose Time to surgery

In Table 3 the mentioned classification methods are compared for the abilities to deal with different data, low sample sizes, high dimensionality, distributed learning and rapid learning.

Table 3. Suitability of classification methods based on type of input data

Classification method	Preferred data type	Dealing with low sample size	Dealing with high dimensionality	Dealing with missing values	Interpretability	Suitable for distributed learning	Suitable to update models rapidly
SVM (3.4.1)	Both	Fair	Good	Fair	Fair	Yes	Yes
RF (3.4.2)	Both	Fair	Good	Fair	Fair	No	Yes
BN (3.4.3)	Categorical	Poor	Poor	Good	Good	Yes	Yes
NN (3.4.4)	Both	Poor	Poor	Fair	Poor	No	Yes

There have been several classifier comparison studies published, also in the genomics domain. Testing 22 diagnostic and prognostic microarray-based datasets by SVMs and RFs showed that random forests are outperformed by support vector machines both in the settings when no gene selection is performed and when several popular gene selection methods are used.[28] On the other hand, random forests are found to be optimal when feature distributions were skewed and when class distributions were unbalanced.[29] Another study found that BNs are outperforming SVMs and RFs when classifying mood disorders based on gene expression and SNP data, but differences in performance are small. [30]

3.5 Clustering

Clustering is a data mining technique that defines groups of observations that have similar characteristics. Contrarily to classification where objects are assigned into predefined classes, clustering both defines the classes and assigns objects to them. By using clustering techniques we can identify particular regions in object space and can discover overall distribution pattern and the correlations among data attributes. Types of clustering methods that we discuss in this deliverable are hierarchical clustering, partitioning clustering, unsupervised random forests, Bayesian clustering and other techniques like coexpression networks, integrative clustering and consensus clustering.

The reasons to do unsupervised clustering:

1. Hypothesis generation
2. Labelling large data sets can be very costly
3. Changes in patterns over time can be detected
4. Data categorization purposes
5. As exploratory phase of data analysis

For EURECA we will use clustering especially for hypothesis generation. For example for the genomic field, suppose genes A and B are grouped in the same cluster, then we hypothesis that genes A and B are involved in similar function. If we know the role of gene A is apoptosis but we do not know if gene B is involved in apoptosis, we can do experiments to confirm if gene B indeed is involved in apoptosis.

The tools for unsupervised learning and clustering are the same as for classification, where R is the main focus for Eureca. For R most clustering tools are summarized and available at: R: http://cran.r-project.org/web/views/Cluster.html

3.5.1 Hierarchical clustering

Hierarchical clustering[31] is based on the core idea of data points being more related to nearby data points than to data points farther away, meaning that these algorithms connect data points to form clusters based on a distance measure. This clustering is called hierarchical because these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. Two types can be distinguished: agglomerative, where each observation start in its own cluster and pairs of clusters are merged as one moves up the hierarchy, and divisive, in which all observations start in one cluster and splits are performed moving down the hierarchy. A useful feature of this type of clustering is the formation of a dendrogram, which visually shows the formation of clusters when the

distance threshold is varied. A published example is provided in Figure 6, showing also the dendrogram and the identified clusters at the top.

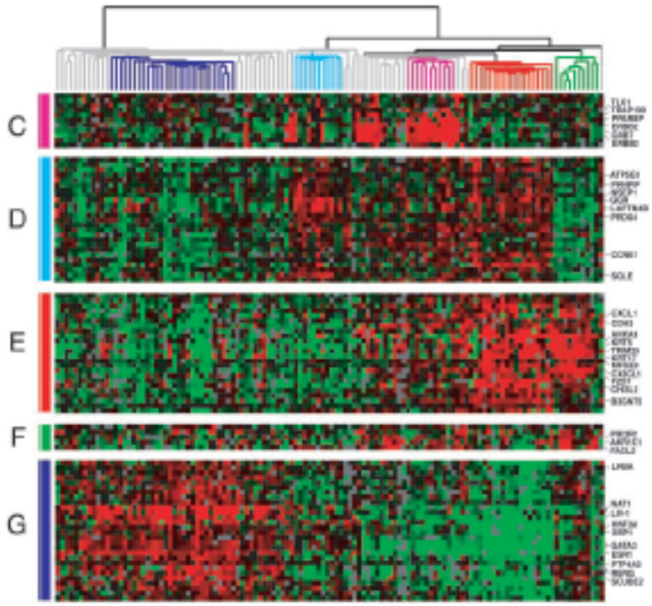

Figure 6. Coexpression map of 534 genes that are clustered hierarchically, identifying five gene clusters in 115 breast cancer tissue samples.[32]

Advantages:

- Number of clusters not required in advance
- No input parameters (except choice of similarity)
- Computes complete hierarchy of clusters
- Integration of result visualizations

Disadvantages:

- Interpretation of the hierarchy is complex
- Only effective at splitting small amounts of data
- Sensitive for outliers because outliers may become their own clusters or will falsely connect distant clusters.
- No automatic discovering of optimal clusters

Published applications:

- Exploring miRNA deregulation and candidate miRNA markers for follicular carcinomas that can be used diagnostically[33]
- Predicting prognosis in colorectal cancer using hierarchical clustering for gene expression[34]

3.5.2 Partitioning clustering

The most well-known and widely used partitioning clustering algorithm is K-means clustering[35]. This method, originating from 1957, aims to partition *n* observations into *k* clusters in which observation belongs to the cluster with the nearest mean. The idea is

to choose random cluster centers, one for each cluster. These centers are preferred to be as far as possible from each other.

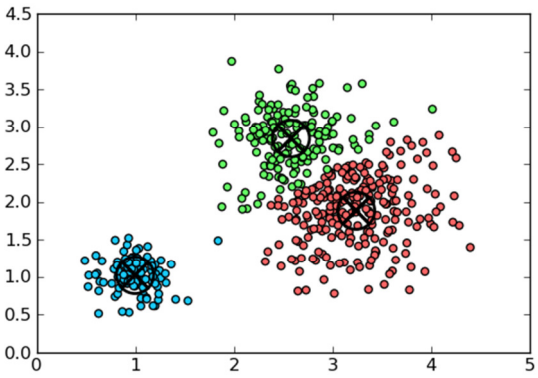

Figure 7. Artificial example of k-means clustering for randomly and normally distributed clusters (k=3)[36]

Advantages:

- Computationally fast for large samples (and for small k)
- Produces tight clusters, especially for globular clusters (convex or spherical/elliptical)

Disadvantages:

- Inappropriate choice of *k* may yield poor results (diagnostic checks are required)
- Sensitive to the randomly chosen initial cluster centres
- The tendency of k-means to produce equivalent sized clusters can lead to counterintuitive and false results
- Might converge to local optimum, resulting in clusters close to the initial partitioning
- Works not well with non-globular clusters

Published applications:

- Weighted K-means clustering for microarray data[37]
- Classification of breast cancer using gene expression, copy number variations and microRNA[38]

3.5.3 Unsupervised random forests

Random forests are usually used for supervised learning, but unsupervised learning is also possible[39]. The approach is to consider the original data as class 1 and to create an artificial second class of the same size that will be labelled as class 2. The artificial second class is created by sampling at random from the univariate distributions of the original data, meaning that class two has the distribution of independent random variables. Class 2 thus destroys the dependency structure in the original data. Now, two classes are created and this two-class problem can be run through random forests. The higher the misclassification rate in this two-class problem is, the input variables are looking too much like independent variables (low discrimination). Other way around, if the misclassification rate is low, the dependencies between the input variables are playing an important role.

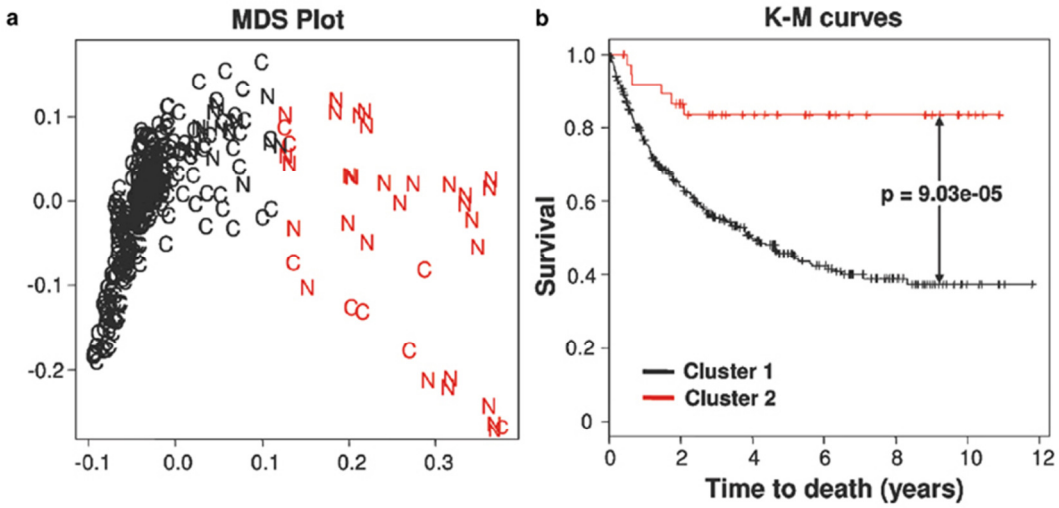

Figure 8. Example of an applied RF clustering method[40]. *a.* multidimensional scaling plot on the RF dissimilarity in the DNA microarray data. Two clusters are defined (C = Clear cell, N = Non-clear cell). *b.* Kaplan-Meier plots of survival show distinct separation between the two clusters.

Advantages:

- Missing values can be replaced effectively
- Outliers can be found
- Scaling can be performed
- Variable importance can be measured
- Can deal with skewed data distributions

Disadvantages:

- Exact quantitative contribution of every variable is hard to interpret.

Published applications:

- Lung tumor classification using supervised and unsupervised random forests.[41]
- Classification for renal cell carcinoma using microarray data (see also Figure 8)[40]

Additional tools:
In R the same function can be used as for the supervised random forests as has been described[42].

3.5.4 Bayesian clustering

Unsupervised Bayesian clustering has been far less explored than the supervised Bayesian networks.[43] Beside the Chow and Lui multinets and the tree augmented Naïve Bayes model, the simple Bayesian network (SBN) classifier has been introduced recently in 2009. This method is more robust in its structure, capable of handling the trade-off between complexity (number of edges in the network) and accuracy using a Bayesian approach. The unsupervised training technique maximizes the classification maximum likelihood (CML) of the Bayesian network classifiers, instead of using the traditional EM approach that maximizes the maximum likelihood (ML). The methods both do structure and parameter learning but without labeling of the outcomes. Not many applications are published of this method, but because in

classification the BN is so elegant and useful, we decided to discuss this clustering technique for comparison.

Advantages:

- Number of clusters is variable
- The resulting structure can give additional information on how the features are related (probabilistic dependencies) in each cluster

Disadvantages:

- It is difficult to determine the correct number of clusters during the learning process
- One needs an effective unsupervised technique for transforming continuous attributes into discrete ones

Applications:
This method has not been used frequently. One paper stated that Bayesian unsupervised learning was useful in finding the dependence between gene expression and micro-RNA data.[44]

3.5.5 Other clustering techniques

3.5.5.1 Co-expression networks

Using gene co-expression analysis clusters of genes with consistent functions that are relevant to cancer development and prognosis can be detected. Gene co-expression networks are constructed from data of gene expression microarray experiments by using different correlation based inference methods.[45] The vertices of these networks represent genes, while their edges are related to the values of the pairwise correlation coefficient that is calculated from the expression data of the genes. Co-expression networks, in contrast with other networks whose edges represent well-defined biological interactions, are composed of edges that show co-expression patterns of genes over different experimental conditions. There are some remarks to be made when interpreting co-expression results. Often linear correlations of the gene expression values are considered, some co-expression edges might be established by simple chance, averaging the gene expression values over a large number of cells could distort the whole co-expression analysis, and the networks are known to be incomplete which may affect the results. Successful applications of this method have been published in which clusters of genes were associated with prognosis[46] or clusters of microRNAs were found to affect disease.[47] An R-package have been developed: http://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/Rpackages/WGCNA/

3.5.5.2 Integrative clustering

When taken into account a genomic dataset involving more than one data type measured in the same set of tumors, for example integrating copy number and gene expression data, it is often called multiple genomic platform (MGP) data. Identifying tumor subtypes by simultaneously analyzing MGP data is a new problem. The current approach to subtype discovery across multiple types is to separately cluster each type and then to manually integrate the results. An ideal integrative clustering approach would allow joint inference from MGP data and generate a single integrated cluster assignment through simultaneously capturing patterns of genomic alterations that are: 1. consistent across multiple data types; 2. specific to individual data types; 3. weak

yet consistent across datasets that would emerge only as a result of combining levels of evidence.[48] A study where this method is applied on a large scale was found for breast cancer in which copy number and gene expression data were integrated.[49] An R implementation for integrative clustering:
http://www.mskcc.org/mskcc/html/85130.cfm

3.5.5.3 Consensus clustering

This strategy involves class discovery and clustering validation tailored to the task of analyzing gene expression data.[50, 51] It refers to the situation in which a number of different (input) clusters have been obtained for a particular dataset and it is desired to find a single (consensus) clustering which is a better fit in some sense than the existing clusters. Many clustering techniques are dependent on the selection of the distance measure and these are hard to define with high dimensionality. Consensus clustering provides a method to represent the consensus across multiple runs of a clustering algorithm, to determine the number of clusters in the data, and to assess the stability of the discovered clusters. A very interesting applications has been published for DNA microarray analysis of lung cancer which could identify reproducible tumor subtypes with different clinically behaviors.[52]

Implementations in R:
http://cran.r-project.org/web/packages/clusterCons/clusterCons.pdf
http://www.bioconductor.org/packages/release/bioc/html/ConsensusClusterPlus.html

3.5.6 Comparison of clustering techniques

Several publications are present about comparing clustering techniques for genetic data specifically, which we will focus on in EURECA for the hypothesis generation scenario. For example, 2780 cluster analysis methods were tested on seven publicly available microarray data sets with common reference designs, and it followed that hierarchical clustering using Ward's method, k-means clustering and Mclust (R: http://www.stat.washington.edu/mclust/) are the clustering methods considered in this paper that achieves the highest adjusted performance.[53] Another publication tested seven different clustering methods for the analysis of 35 cancer gene expression data sets.[54] Here, finite mixture of Gaussians[55] performed best, followed closely by k-means. It is a difficult task to select a cluster algorithm which is optimal for your dataset within the huge list of methods currently developed. This selection process can also be formalized by validation measures. A publication aiming for these measures defined one measuring the statistical stability of the clusters produced and another one representing their biological functional congruence.[56]

3.6 Text mining

Text Mining is a branch of data mining that refers to learning by using automatic extraction of information from free text. Information from different text documents and/or resources is extracted and then linked to generate new rules or hypotheses. These are typically organized and explored with other data-mining methods. In text mining the data patterns are extracted from natural language text rather than from structured databases; such automated processing of natural language is challenging and methods to perform such a task are still limited. Typically, it requires dividing text mining in specific relatively small tasks that can be performed automatically.

An example of application to genomics is the study of co-occurrences of words in publications to infer related function of genes or proteins, or generate hypotheses that can then be tested in further studies.[57]

For further information on text mining in EURECA we would like to refer to deliverable 3.1, where the SPECIALIST NLP Tools and Medical text processing toolkit are described.

3.7 Time-dependent and data stream mining

In many modern scientific and medical research domains, new knowledge and data are stored and recorded in large data streams of transactional data that rapidly and continuously grow over time. Not all scenarios described in section 4 present all the data stream characteristics but several of them present the challenge of being data that change over time and a large amount of data to be processed. These types of data require a different data mining approach with respect to the ones used for classical static databases; both as the data change in time but also because the dimension of the data does not allow the classical re-sampling and training approaches often used in the machine learning and data mining community. For example, several classification algorithms require a recursive processing of the data.

Methods for analysis of such data have been described and a recent collection has been published, edited by Aggarwal, which describes many of the advances in the area [58].

3.8 Privacy preserving data mining

One of EURECA key goals is to deliver an environment that fulfils the data protection and security needs and the legal, ethical and regulatory requirements related to linking research and EHR data. In addition, several conflicting interests of different stakeholders must be taken into account to ensure the practicability of data mining solutions. The main problem are conflicting interests on which information should be protected, and which information should be freely available, in particular when considering information that should be made public outside of the EURECA contractual framework, e.g. in the form of scientific publications or open models for decision support. Vast research on methods for privacy-preserving data mining exists. The main directions of privacy-preserving data mining can be described as follows:

Privacy-preserving data publishing deals with the question of releasing data in such a way that all sensitive information is removed. The released data can then be processed with standard data mining methods. Approaches include randomization, k-anonymity [59], l-diversity [60] and t-closeness [61]. In EURECA, however, these approaches do not seem promising. The problem is that once relevant information is removed, it cannot be recovered. Strategies for EURECA specific privacy preserving data mining are explained in detail in deliverable 5.1.

3.9 Distributed data learning

A vast amount of information is currently stored in digital data repositories, yet it is often difficult to understand and extract the important and useful information in those

massive data sets. To sift large data sources, computer scientists designed software techniques and tools that can analyse data to find useful patterns—these techniques contribute to the so-called knowledge discovery in databases (KDD) process. In particular, data mining is the basic component of the KDD process for the semiautomatic discovery of patterns, associations, changes, anomalies, events, and semantically significant structures in data. Typical examples of data mining tasks are data classification and clustering, events and values prediction, association rules discovery.

Cloud and grid computing are the most promising frameworks for future implementations of high-performance data-intensive distributed applications. Furthermore, the Internet is shifting from an information and communication infrastructure to a knowledge delivery infrastructure. The discovery and extraction of knowledge from geographically distributed sources will be increasingly important in many typical daily activities. The distributed knowledge discover is a significant step in the process of studying the unification of knowledge discovery technologies and defining an integrating architecture for distributed data mining and knowledge discovery based on cloud or grid services. Such architectures will accelerate progress for very large-scale geographically distributed data mining by enabling the integration of various currently disjointed approaches and revealing technology gaps requiring further research and development.

The basic principles behind the reference architecture design of a distributed KDD system include: *Data heterogeneity and large data-set-handling; Algorithm integration and independence; Compatibility with distributed infrastructure; Openness; Scalability; and Security and data privacy*.

3.9.1 Distributed data mining tools

EBI-R-Cloud (http://www.ebi.ac.uk/Tools/rcloud/)**.** EBI-R-Cloud is a new service at the European Bioinformatics Institute (EBI) allowing advanced users of the statistical package R to log on and run distributed computational jobs remotely, making use of the powerful EBI infrastructure. Users log on to the system and can work on multiple projects, submitting long-running, memory-intensive tasks that make use of multiple computational nodes. EBI-R-Cloud comes with a full mirror of CRAN and Bioconductor package repositories. Users have access to all public data hosted at the EBI without the need to download it to their machines. EBI-R-Cloud also includes the **R-Cloud-Workbench** (Mac OS and Windows versions), an optimized graphical client to R on the cloud.

The EBI-R-Cloud showcase include three distributed computing applications: (a) distributed Affymetrix microarray data normalization; (b) genotype imputation in the cloud, and (c) ArrayExpressHTS, distributed pre-processing and quality assessment of RNA-seq datasets – with Data from the 1000 genomes are available as a reference panel option.

Bioconductor AMI (http://www.bioconductor.org/help/bioconductor-cloud-ami/). An Amazon Machine Image (AMI) is developed and optimized for running Bioconductor in the Amazon Elastic Compute Cloud (or EC2) for sequencing tasks when: one does not want to install Bioconductor on her own machine; a long-running task that may tie up

the CPU; one has a parallelizable task and would like to run it (either on multiple CPUs on a single machine, or in a cluster of many machines); run R-Bioconductor on a web browser (using RStudio Server; or run difficult to install and configure packages like RGraphviz)

GridR (http://cran.r-project.org/web/packages/GridR/index.html). GridR is a tool which allows using the collection of methodologies available as R packages in a grid environment. The aim of GridR, which was initiated in the context of the ACGT EU project, is to provide a powerful framework for the analysis of clinico-genomic trials involving large amount of data (e.g. microarray-based clinical trials). As a proof of concept, an example of microarray-based analysis taken from the literature was reproduced using GridR. GridR is an R-Package that submits R functions to execute them on another computer or cluster and it provides an interface to share functions and variables with other users. Submission modes are using a web service, ssh or local, execution modes are condor, globus or using a single server. All needed functions and variables that are necessary to execute that function will be copied to the execution machine.

3.10 Subgroup Discovery

Subgroup discovery is a technique for learning descriptive rules, i.e. rules that can be used to understand inherent relations in the data of a database. A subgroup is a subset of individuals in the database such that the individuals in the subgroup are distinguished from all other individuals by their characteristics with regard to some target attribute. Typically, these characteristics ensure that a subgroup displays a different (statistical) distribution on the target attribute if compared to the distribution on the target attribute in the complete dataset. Subgroups are presented in terms of "subgroup patterns", i.e. a conjunction of atomic propositions, the most common being pairs $a = v$ with a being an attribute of the database and v being a value. Then a subgroup pattern of the form $a_0 = v_0, \ldots, a_{n-1} = v_{n-1}$ states that the combination of values v_i of the attributes a_i are "interesting" with regard to a specific attribute studied.

In case of numerical values, atomic propositions may be intervals $l \leq a \leq u$ with l being a lower and u being an upper bound. An instance d_i of the database d is said to satisfy a subgroup pattern p if it satisfies all the atomic propositions in the pattern, e.g.. if $a = v$ is in p, then $d_i(a) = v$ meaning that the value $d_i(a)$ of the instance d_i at attribute has value v.

Subgroup discovery intends to find subgroup patterns that are "interesting" and "easy to interpret". Interestingness is typically expressed in terms of a "quality function" that typically reflects statistical or other user-defined criteria. Moreover, subgroup patterns found are often relatively simple and thus easy to understand. For example, if biomedical experts try to identify which genes may play a role in the development of a disease, gene expressions of each patient's DNA can be related to the relevant clinical data, with particular clinical data being used as target attribute.

Subgroup discovery is a generic name for a variety of algorithms with specific characteristics, for instance

- Type of values of target attribute

  - Nominal
  - Numeric
  - Ordinal
- Subgroup pattern
  - Nominal - $a = v$
  - Intervals - $l \leq a \leq u$. The intervals may be disjoint or overlapping. There are several strategies to define intervals for specific data, e.g. divide data into intervals that hold equal number of values or split the range of all values of an attribute into equal length intervals.
- Quality function
  - Two class quality functions – the values of the target attribute are split into two classes in terms of which the quality function is defined.
  - Multiclass quality functions – the quality function depends on all values of the target attribute.
  - Exceptional Mode Mining – the quality function depends on a complex statistical property of the examples covered by the subgroup
- Subgroup property
  - Refined subgroups – only maximal elements with regard to a given order on subgroups are considered for the output
  - Closed subgroups – close a subgroup pattern under all patterns that have the same "instance basis", i.e. which occur in all the instances in which the given subgroup pattern occurs.

Each particular choice of algorithm reflects particular aspects of "interestingness" and results in quite different subgroup patterns being generated. Hence working on a use case, subgroup discovery implies exploration of the "space of algorithm". Thus, the choice of algorithms becomes a parameter for subgroup discovery. Other parameters include the length of subgroups to be considered (which may have a dramatic effect concerning computation time and space), number of best subgroups to be considered, minimal quality, and generality.

3.10.1 Subgroup Discovery for Genomic Data Analysis

The main purpose of a typical microarray experiment is to find a molecular explanation for a given macroscopic observation. The most common methods are based on a 'functional enrichment'. First, genes of interest (e.g. genes that are significantly over- or under expressed when two classes of experiments are compared) are selected. Then, external sources of information, such as gene ontologies and pathways databases, are included to translate the set of genes into interpretable biological knowledge. SD can be extended as an approach that transforms the dataset submitted by the user into a large list of genes enriched by GO terms, see Trajkovsky[62].

Applied to gene expression data, the standard SD algorithm would deliver a set of gene names that share similar properties relative to the research question of interest, i.e. SD uses the filtered dataset as produced by the statistical methodology used. The translation of these results into useful biological knowledge still remains a necessary validation procedure, which is often time-consuming. For instance, one might wonder how the set of genes can be described in terms of molecular or cellular function. Knowledge databases such as Gene Ontology[63] (GO) or Kyoto Encyclopedia of Genes and Genomes[64] (KEGG) serve as an excellent basis for the interpretation of genes.

Gene Ontology (GO) serves as a controlled vocabulary of terms for describing genes according to several aspects. GO includes three ontologies containing the description of molecular functions, biological processes and cellular locations of any gene product, respectively. Within each of these ontologies, the terms are organised in a hierarchical way, according to parent-child relationships in a directed acyclic graph (DAG). This allows a progressive functional description, matching the current level of experimental characterization of the corresponding gene product. Moreover, further interesting knowledge databases exist, such as the KEGG or Reactom[65].

Overall, the integration of these additional knowledge databases into Subgroup Discovery would provide the researcher with more meaningful results.

4 Technical scenarios using DM

In this section an overview of all the technical scenarios which use or may use data mining methods is provided. Other scenarios are not listed here for clarity purposes.

4.1 Personal Medical Information Recommender

Partner scenarios: SIT2, VUA2, UdS3
Responsible partner: StoneRoos

Description

The personal medical information recommender (PMIR) is a tool reporting to either the patient (health condition, EHR assistance) and physicians (e.g. relevant literature). The recommendations are generated by the local PMIR Service which applies several types of approaches to find the relevant contextualized semantic match between the personal health information that originates from the local EHR-DW and CDW accessible via the CIM based Data Access layer and the remote PMIR Metadata Service (PMS). The PMS updates periodically the meta-data that it extracted from the external sources that are registered by the subscribed administrators of the PMS.

Data mining

In the current status of the scenario it is unknown if the involved recommender algorithm will use data mining techniques. It is expected that use cases UC.TS.IR.02 (reporting relevant literature) and UC.TS.IR.04 (update relevant patient information) will be candidates to use data mining techniques because each extraction task involves three parallel processes getting recommendations about terminology, sources and literature. Several studies have been published who developed a grading scale for medical evidence based on specific literature. For example, a scale that allows readers to learn one taxonomy that will apply to many sources of evidence based on quality, quantity, and consistency of the evidence, allowing authors to rate individual studies or bodies of evidence.[66]

4.2 Data mining for consultation

Partner scenarios: UdS2
Responsible partner: FhG IAIS

Description

The goal of this scenario is to help a trial chairman to answer frequently asked questions in consultations posed by clinicians. It involves entering and viewing consultation requests and replies and giving feedback on consultations.

Data mining

The consultation tool will computes the similarities between the current CRF and all CRFs in the system when viewing a consultation recommendation (UC.CD.CR.2). Currently it is not known if this involves similarity learning, as has been described in section 3.2.

4.3 Update of guidelines

Partner scenarios: UdS4, Maastro1
Responsible partner: VUA

Description
Support the updating process of a guideline by identifying relevant literature (evidence) for this guideline.

Data mining
The guideline developer is able to find new and relevant evidences from papers in PubMed or clinical trial repositories based on the evidences of a guideline. This is based on a set of keywords from the evidence description, keywords of the papers, and the references of the papers to identify the relevance. The system shows the potentially relevant literature. It is expected to use DM methods to calculate the relevance score to rank literature.

Automatic grading systems for medical evidence have been developed using machine learning techniques, obtaining 70% accuracy using publication types, publication years, journal information and article titles.[67] Others computed relevance and quality scores to rank the literature according to their evidence.[68] An annotation scheme based on an evidence-based medicine model for critical appraisal of evidence was developed. Textual, structural, and meta-information features essential to outcome identification were learned from the created collection and used to develop an automatic system. Accuracy of automatic outcome identification was assessed in an intrinsic evaluation and in an extrinsic evaluation, in which ranking of MEDLINE search results obtained using PubMed Clinical Queries relied on identified outcome statements. It is also possible to find specific outcome statements by evaluating each sentence in medical text (micro approach). In contrast, classifying a text as a whole to determine the presence of an outcome statement, showed only moderate associations between perceived clinical value of a citation and features that characterize the whole citation (macro-level approach).[69] Another approach is to construct a rule-based tree to calculate stability and strength of medical evidence (Figure 9).

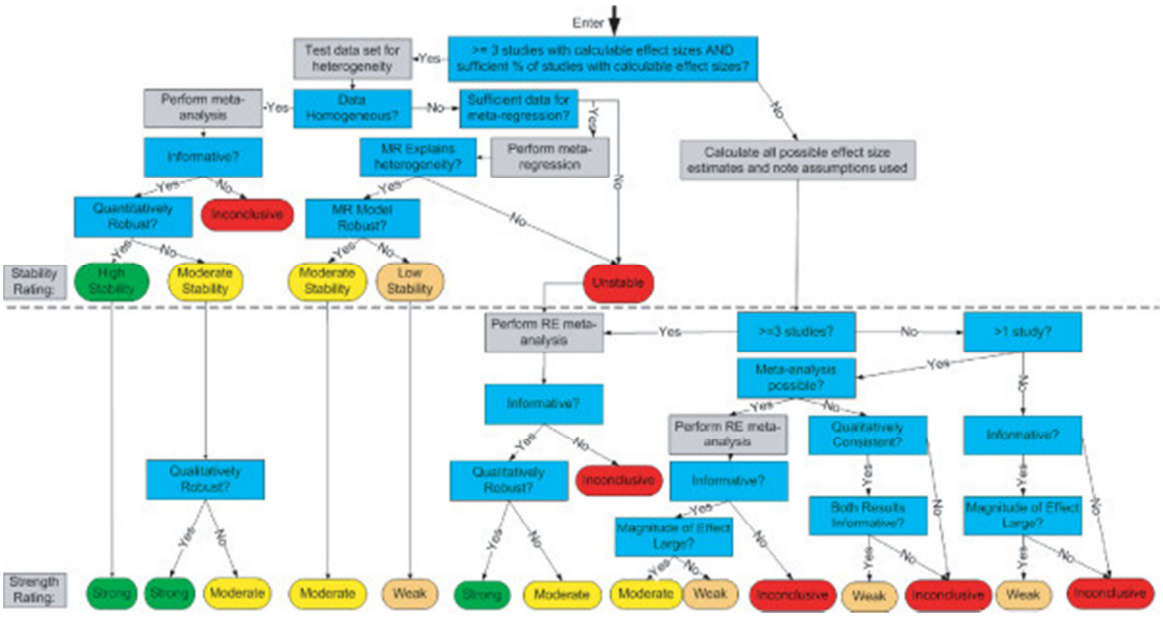

Figure 9. Example of rule-based tree to calculate stability and strength of medical evidence.[70]

4.4 Training/validating/updating a diagnostic classifier

Partner scenarios: UOXF1, Maastro4
Responsible partner: UOXF

Description
This scenario describes the process of obtaining a diagnostic classifier by using a set of data mining tools on the collected patient health information including genomic data, images and clinical data that are gathered from EPRs. It involves pulling the data from the COSD repository, pre-processing the data for the training process, configuring and run the training tools to obtain the classifier, selecting a trained diagnostic classifier, loading patient health information and running the classifier to obtain classification results.

Data mining
In this scenario data mining is particularly used in the training phase of the classifier, whether this is a first training or an update. The aim is to learn a diagnostic classifier from clinical, biomarker, imaging and genomic data, and to then use this classifier to improve the management of the patients and choosing optimal treatment. This scenario will use mainly supervised methods on the combined analysis of genomic data and clinical data. More specifically, the data-mining will mainly involve classification in early phase trials and routine care patients, and will generate a classifier which will be refined in subsequent large scale studies.

The scenario requires no de-identification if it runs locally exclusively inside the hospital walls but whenever this is not possible de-identification is required. Semantic operability needs to be harmonized to use the tool in different centres. Data involved are clinical, biomarker, treatment, imaging and genomic related. In this context, the model will need to be updated and tool such as rapid learning will be used, in which the classifier is learnt first on specific set of patients from trials and routine care and then continuously updated and validated with the available data from routine patient care. The rapid learning concept has been described in a previous document (D5.1 “Requirements analysis and knowledge discovery scenario”, in section 4.2). This approach is extremely useful in the context of a diagnostic classifier because very limited data, both clinical and mostly genomic data, is currently available for the validation of the diagnostic classifier. Distributed learning will also facilitate the modelling process because large numbers of data are required to accurately train the models.

4.5 Hypothesis generation

Partner scenarios: UdS6
Responsible partner: UOXF

Description
This scenario involves the support in designing new trials and hypothesis generation, which allows the clinical and scientific researcher users to generate new hypotheses from existing clinical trials, literature, public databases and experimental data in an

efficient, and automatic or semi-automatic way. A trial design is then suggested that is suitable to test the new hypothesis. The steps that are involved are:

- Pull data from clinical trials, literature, public databases and other available experimental evidence
- Mine the data to generate multiple hypotheses
- Finalize the new hypothesis to test in clinical trials/studies
- Identify possible designs for clinical trial/study and check feasibility

Data mining

The background and more details for this scenario, and an example in the genomic area have been already provided in a previous deliverable (D5.1 “Requirements analysis and knowledge discovery scenario”, in section 4.3).

Briefly, a clinical trial often starts with the formulation of a research hypothesis generated as a consequence of analysing available data from previous trials, guidelines, existing literature and/or laboratory results. Depending on the validation of the hypothesis and evidence already acquired an early or late phase trial will be designed. A trial can often also involve a translational biomarker study, where biomarker can be tested during the trial execution. However is often the case that the biomarker study is planned after the execution of the trial has completed, limiting the scope of the biomarker study. An early and fast knowledge management and hypothesis generation could help this process and result in better designed clinical trials and biomarker or in general translational studies.

In EURECA, this task will be applied mainly to the genomic area and the use of previously acquired genomic data to formulate hypotheses to be tested in and to help the design of clinical trials, pharmacodynamics studies, translational genomic studies and biomarker studies. This scenario will use both supervised and unsupervised methods on the combined analysis of genomic data and clinical data. More specifically, the data-mining will mainly involve classification and clustering in retrospective clinical series and early phases trials and will generate hypotheses to design subsequent larger trials and associated translational studies.

The scenario requires no de-identification if it runs locally exclusively inside the hospital walls but whenever this is not possible de-identification is required. Semantic operability needs to be harmonized to use the tool in different centres. Data involved are clinical, treatment related, imaging, blood biomarkers, genomic related.

This scenario also makes use of the rapid learning tool, in which hypothesis generation is continuously refined with new evidence and validated with the available data from clinical trials, translational studies and routine patient care. The rapid learning concept has been described in a previous document (D5.1 “Requirements analysis and knowledge discovery scenario”, in section 4.2). In fact, due to the fast accumulation of scientific, clinical knowledge, and an interactive expertise component, this scenario would benefit from some of the methods used in data stream mining, where mining is performed over continuously and rapidly changing streams of data (see D5.1 “Requirements analysis and knowledge discovery scenario”, Section 3.1.1 and also this document).

4.6 Protocol feasibility

Partner scenarios: VUA3, UdS7
Responsible partner: Philips

Description
In this scenario the aims are to define a trial proposal, request an evaluation of the recruitment potential of a trial for selected data sources, and view the results of the evaluation.

Data mining
The first use case expected to use DM techniques is UC.TS.PF.11 (compute eligibility criterion probability). Here the researcher can select sources of public data which will be used to automatically determine the probability of a successful outcome of the criterion. Another candidate use case is UC.TS.PF.13 (compute trial path probability). The researcher can select sources of public data which will be used to automatically determine the probabilities, thereby modelling the distribution of the different trial paths.

4.7 Microbiology SAE

Partner scenarios: UdS8
Responsible partner: FhG IBMT

Description
In the scenario first the trial chairman defines in one or more specific CRFs which specific information have to be documented in order to get an early knowledge about infectious agents and their resistance profile for patients in a chemotherapy. Services enable the collection of data from the Microbiology database for a specific patient in order to get specific information as defined in the CRFs of the Microbiology Module. These data will be automatically included in the corresponding CRF. As far as Common Toxicity Criteria are defined, a SAE event can be automatically detected. Services also enable the collection of data from the Hospital Information system (HIS) for a specific patient in order to get specific information as defined in the CRFs of the Microbiology Module. These data will be automatically included in the corresponding CRF. As far as Common Toxicity Criteria are defined, a SAE event can be automatically detected and reported.

Data mining
Use case UC.TS.MS.05 (statistical analyses of specific infection/medication based parameters) involves only the interface to export data. Exported statistical parameters can be:

- Summary of the SAE of the patient
- Summary of all infections of a specific patient with all infectious agents, their source and resistance profile, usage of antibiotics for each infectious disease
- Summary of infectious agents, their source and resistance profile of a ward, or of a specific infection (e.g.: pneumonia) and a list of antibiotics used
- Comparison of the above generated data with other oncology wards or other wards in the same hospital or outside.

The main end task of this scenario is to detect SAEs, which is a classification task. More on serious adverse events can be found in section 4.9.

4.8 Outcome prediction

Partner scenarios: Maastro2, Maastro3
Responsible partner: UOXF

Description
This scenario describes the process of making predictions for patient outcome after treatment in order to assist physicians in making treatment decisions for new patients. This assistance involves tools to train and update prediction models within the rapid learning framework, and to translate these models to a validated a decision support system based on all the available data including genomic data, images and clinical data that are gathered from EPRs.

Data mining
In EURECA, the outcome prediction mainly involves feature selection, classification, distributed learning. The involved scenarios describe why it is useful to do outcome prediction; physician and patient require an estimate of the outcome for certain treatments, in order to make a decision which treatment fits the patient wishes and benefits best. The involved tool uses existing, validated outcome prediction models and allows this kind of decision support. The outcomes that are predicted are cancer specific, but generally tumour response, local control, distant disease, survival, quality of life, cost and toxicities. It is shown before that outcome predictions by doctors are of low accuracy.[71]

The tool requires no de-identification if it runs locally in the hospitals but for all instances of the model running externally de-identification is required. Semantic operability needs to be harmonized to use the tool in all centres. Data involved are clinical, treatment related, imaging, blood biomarkers, genomic related.

The basis of these outcome prediction tools is the rapid learning tool, in which outcome prediction models are continuously learned and validated with the available data from routine patient care. The rapid learning concept has been described in a previous document (D5.1 “Requirements analysis and knowledge discovery scenario”, in section 4.2). This approach is extremely useful in the context of outcome prediction because limited data is currently available for model validation. Distributed learning will facilitate the modelling process because large numbers of data are required to accurately train the models. The aim is to develop models that have sufficient discrimination (accuracy to distinguish outcomes with the model) and calibration (measure for how good the model output distribution represents the real outcome distribution) and optionally prediction intervals to provide a confidence measure.

4.9 Automatic detection and reporting of SAEs/SUSARs

Partner scenarios: Maastro8, UdS10, UdS11
Responsible partner: FhG IBMT

Description

This scenario involves the automatic detection of SAEs and SUSARs, which will be (automatically) reported through a specific CRF in the clinical trial system.

Data mining
In use case UC.CD.AD.03 (Detection of a SAE) the clinical trial system detects a SAE automatically based on defined criteria (UC.CD.AD.01) or/and through data mining processing. Previously, automatic detection of adverse drug reactions (ADR) have been published using natural-language processing (NLP) and a knowledge source to differentiate cases in which the patient's disease is responsible for the event rather than a drug.[72-74] Accuracy and time-saving are important outcome targets here.

4.10 Economic analysis

Partner scenarios: UdS13
Responsible partner: FhG IBMT

Description
The scenario describes data mining of hospital data for economic purposes. The goal is to help hospital administration in a better understanding of the economic effects of patient treatment. Steps involved are: selecting data to analyse, joining additional information, select and execute analysis, view results.

Data mining
Use case UC.NN.NN.3 (select analysis) can be affected by data mining. Tools involve data mining of hospital data for economic purposes. A list of possible analyses that can be performed on the selected data is shown to user.

Papers have been published about the mining EHRs with the purpose to reduce healthcare costs or improve management. To analyse hospitalized patient flows one might use sequential pattern mining, very values are delivered in a sequence as is the case with hospitalization.[75] A framework tool for sequential pattern mining is available (http://www.philippe-fournier-viger.com/spmf/). Association and classification rule mining can also be used for this surveillance purpose.[76] A showcase of several methods implemented in SAS Enterprise miner (http://www.sas.com/technologies/analytics/datamining/miner/) is also available (http://www2.sas.com/proceedings/sugi31/077-31.pdf)

5 DATA MINING PLATFORMS/TOOLS

There are many tools available for data mining if you consider also the numerical computing environments :

- SPSS (http://www-01.ibm.com/software/uk/analytics/spss/)
- R (http://www.r-project.org/)
- Weka (http://www.cs.waikato.ac.nz/ml/weka/)
- Matlab (http://www.mathworks.co.uk/products/matlab/)
- Splus (http://www.morningstarcommodity.com/get-support/downloads/statistical-tools/s-plus-download)
- Mathematica (http://www.wolfram.co.uk/mathematica/)

We do not want to elaborate on all of these methods and only focus on relevant platforms for EURECA. We can restrict the number of platforms if we consider the following criteria:

- Open source
- Free
- Large community (computational, biomedical, statistical), implying lots of documentation and specific packages and toolboxes.

When these selection criteria are applied, the main driving platform will be R, which is an open source free environment with a large statistics and bioinformatics community. All the other platforms mentioned (except for Weka) are commercial products requiring licenses, which is not preferred in EURECA. Therefore, we will focus on both R and Weka, which has integration in R.

5.1 R platform

R (http://www.r-project.org/) is a free software environment using a free software language for statistical computing. Large statistical and data mining communities are using the environment and contributing to it by developing statistical and data analysis packages. R uses the S programming language (http://cm.bell-labs.com/cm/ms/departments/sia/S/history.html) combined with lexical scoping semantics inspired by Scheme. R was created by Ross Ihaka and Robert Gentleman (the name R is based on their first names) at the University of Auckland, New Zealand. The source code for the R software environment is written primarily in C, Fortran, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. R uses a command line interface; however, several graphical user interfaces are available for use with R.

Positive aspects of R:

- R is a programming environment well suited for statistical analysis.
- R is open source and cross platforms (Windows, Mac, Linux).
- Fortran, C (C++), and Python wrappers are in place.
- Deals well with spatial data, has a robust graphical interface and has an active user group list / forum.
- External packages for R are almost daily increasing, most of them based on published up-to-date books and peer-reviewed articles.
- Documentation

Negative aspects of R:

- R has a steep learning curve.
- Experience with other programming languages is a plus / minus.
- You can save scripts, but not *.exe.
- It is updated several times a year (good) but there are no upgrades.
- Memory management problems (depends on your OS), especially when displaying big images at high resolution or working with huge matrices (hundreds of Mb).

It is very efficient to use a graphical user interface for R, since it only comes with a command line. There are many interfaces available, but here are some notable ones:

- R-studio (http://www.rstudio.com/), see Figure 10
- RGUI (http://www.sciviews.org/_rgui/)
- RapidMiner (http://rapid-i.com/content/view/181/190/)

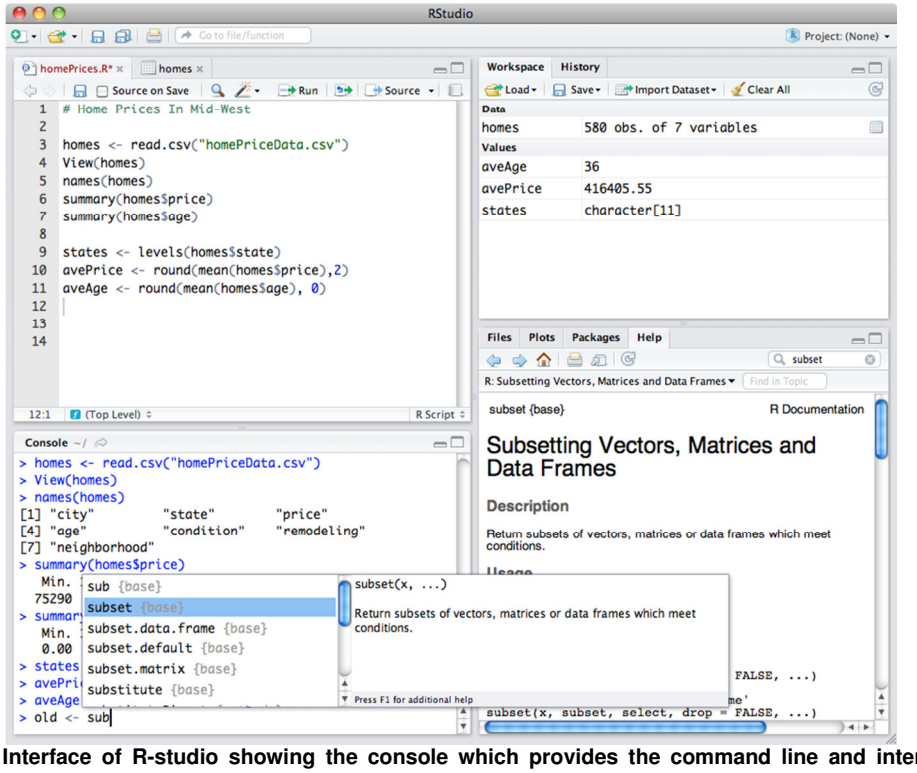

Figure 10. Interface of R-studio showing the console which provides the command line and interfaces for workspaces, history backup, function writing, file explorer, packages (with easy installing features), plotting and help function.

A specific open-source software for the analysis of genomic data using R programming language (but also other languages) is bioconductor (http://www.bioconductor.org/). Many bioconductor packages are available for R and free to download.

Some suggest that EHR vendors will soon be embedding or otherwise implementing R in their solutions, because data mining and analysis of electronic medical record data is the next frontier.[7] R is suitable for this because:

- The used platform should be flexible and capable of adapting to shifting EMR standards. R is positioned well for this environment, as it already integrates and connects into a plethora of database management systems.

- R enables parallel processing and can be used in conjunction with Hadoop (http://hadoop.apache.org/) and other technologies to spread analysis out to distributed hardware.
- The technology will need to be capable of analysing very large amounts of data. As the EHR space is a rapidly growing field, the analytical technology that it's paired with should also be on a growth trajectory. Given that R is open-source, new methods and techniques are implemented into R faster than proprietary alternatives.
- The analytical technology should work on many different operating systems in order to service the variety of hardware/software solutions used by healthcare organizations. R works on Windows, Mac, and Unix.
- The analytical technology should have a large user base to support the needs of the healthcare space. R has a large, international community that includes some of the brightest minds.
- The technology must be transparent. Once again, R is open-source, enabling anyone to go in and understand what it is doing. Also, R is very well-documented in the literature.
- The technology must have very strong support for unstructured data analysis, as much of EHR data is unstructured text. R has a list of very powerful text mining and unstructured data analysis packages / libraries.

5.2 WEKA

Weka (http://www.cs.waikato.ac.nz/ml/weka/) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka[77] is free software available under the GNU General Public License. Weka supports several standard data mining tasks, more specifically, data pre-processing, clustering, classification, regression, visualization, and feature selection. Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation (with the extension ".arff"), where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka's main user interface is the *Explorer*, but essentially the same functionality can be accessed through the component-based *Knowledge Flow* interface and from the command line. There is also the *Experimenter*, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets (see Figure 11**Error! Reference source not found.**).

Figure 11. The Weka machine-learning and data-mining benchmark.

The *Explorer* interface features several panels providing access to the main components of the workbench:

- The *Pre-process* panel has facilities for importing data from a database, a CSV file, etc., and for pre-processing this data using a so-called *filtering* algorithm. These filters can be used to transform the data (e.g., turning numeric attributes into discrete ones) and make it possible to delete instances and attributes according to specific criteria.
- The *Classify* panel enables the user to apply classification and regression algorithms (indiscriminately called *classifiers* in Weka) to the resulting dataset, to estimate the accuracy of the resulting predictive model, and to visualize erroneous predictions, ROC curves, etc., or the model itself (if the model is amenable to visualization like, e.g., a decision tree).
- The *Associate* panel provides access to association rule learners that attempt to identify all important interrelationships between attributes in the data.
- The *Cluster* panel gives access to the clustering techniques in Weka, e.g., the simple k-means algorithm. There is also an implementation of the expectation maximization algorithm for learning a mixture of normal distributions.
- The *Select attributes* panel provides algorithms for identifying the most predictive attributes in a dataset.
- The *Visualize* panel shows a scatter plot matrix, where individual scatter plots can be selected and enlarged, and analysed further using various selection operators.

Positive aspects of WEKA:

- Open source (free, extensible, can be integrated into other java packages)
- Graphic User Interface (easy to use)
- Features: run individual experiment or build knowledge discovery and data mining phases
- Integration with R possible

Negative aspects of WEKA:

- Lack of proper and adequate documentations
- Systems are updated constantly (scope creep)

Weka for EURECA

Even though the core knowledge discovery platform for EURECA will be the R-statistics platform, Weka can be used within EURECA as a complimentary library of on machine learning algorithms for data mining tasks. Linking R-statistics and Weka can be two fold;

- Using a specific R interface to use Weka from the R-statistics environment called RWeka[78]. Package RWeka contains the interface code, and the Weka jar for the machine learning algorithms included in Weka such as data pre-processing, classification, regression, clustering and association rules.

- Weka has a package that brings the power of R into the Weka framework, called RPlugin. The plugin provides a Knowledge Flow component for executing an R script and a wrapper classifier for the MLR (machine learning in R) R package. Also provides a Knowledge Flow perspective and Explorer plugin that implements an interactive R console and allows visualization of graphics produced by R.

6 Summary

Concluding this review of methods and tools for data mining in EURECA, it is recognized that not many scenarios will use data mining. Hypothesis generation, outcome prediction and diagnostic classifier heavily rely on data mining techniques and state-of-the-art methods for them are reviewed in this deliverable. Other scenarios mainly use text mining or similarity learning, but in this stage of the project it is not always known if data mining methods will be used for some scenarios. In those cases this review suggests methods and references for consideration.

7 REFERENCES

1. Rendell KK. The Feature Selection Problem: Traditional Methods and a New Algorithm. 10th International Conference on Artificial Intelligence. 1992: 129-34.
2. Langley P. Selection of relevant features in machine learning. AAAI Fall Symposium on Relevance; 1994; 1994. p. 140–4.
3. M. Dash aHL. Feature selection methods for classifications. Intelligent Data Analysis. 1997: vol. 1, no. 3.
4. A.L. Blum aRLR. Training a 3-node neural networks is NP-complete. Neural Networks. 1992 vol. 5, no. 1:117-27.
5. Fukunaga PMNaK. A branch and bound algorithm for feature subset se-lection. IEEE Trans on Computers. 1977 vol. C-26, no. 9:917-22.
6. Bell RMK, Y.; Volinksy, C. The BellKor solution to the Netflix Prize. 2007.
7. [cited; Available from: http://reference.wolfram.com/mathematica/guide/DistanceAndSimilarityMeasures.html
8. Friesen N, Ruping S. Distance Metric Learning for Recommender Systems in Complex Domains Mastering Data-Intensive Collaboration through the Synergy of Human and Machine Reasoning. CSCW 2012. Seattle, WA; 2012.
9. Agarwal R, Aggarwal C, Prasad V. A tree projection algorithm for generation of frequent itemsets. High Performance Data Mining Workshop; 1999; Puerto Rico; 1999.
10. Aggarwal C, Wof J, Yu P. A new method for similarity indexing for market data. ACM SIGMOD Conference; 1999; 1999.
11. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. ACM SIGMOD International Conference on Management of Data; 1993; 1993.
12. Agrawal R, Srikant R. Fast algorithms for mining association rules. 20th Int Conference on Very Large Data Bases, VLDB94; 1994; 1994.
13. del Jesus MJ, Games JA, González P, Puerta JM. On the discovery of association rules by means of evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2011; **1**(5).
14. Creighton C, Hanash S. Mining gene expression databases for association rules. Bioinformatics. 2003; **19**(1): 79-86.
15. Carmona-Saez P, Chagoyen M, Rodriguez A, Trelles O, Carazo JM, Pascual-Montano A. Integrated analysis of gene expression by Association Rules Discovery. BMC Bioinformatics. 2006; **7**: 54.
16. Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995; **20**(3): 273-97.
17. Patnaik SK, Yendamuri S, Kannisto E, Kucharczuk JC, Singhal S, Vachani A. MicroRNA expression profiles of whole blood in lung adenocarcinoma. PLoS One. 2012; **7**(9): e46045.
18. Smeets A, Daemen A, Vanden Bempt I, Gevaert O, Claes B, Wildiers H, et al. Prediction of lymph node involvement in breast cancer from primary tumor tissue using gene expression profiling and miRNAs. Breast Cancer Res Treat. 2011; **129**(3): 767-76.
19. Schulte JH, Schowe B, Mestdagh P, Kaderali L, Kalaghatgi P, Schlierf S, et al. Accurate prediction of neuroblastoma outcome based on miRNA expression profiles. Int J Cancer. 2010; **127**(10): 2374-85.
20. Soinov LA, Krestyaninova MA, Brazma A. Towards reconstruction of gene networks from expression data by supervised learning. Genome Biol. 2003; **4**(1): R6.

21. Lazar V, Ecsedi S, Vizkeleti L, Rakosy Z, Boross G, Szappanos B, et al. Marked genetic differences between BRAF and NRAS mutated primary melanomas as revealed by array comparative genomic hybridization. Melanoma Res. 2012; **22**(3): 202-14.
22. Bray I, Bryan K, Prenter S, Buckley PG, Foley NH, Murphy DM, et al. Widespread dysregulation of MiRNAs by MYCN amplification and chromosomal imbalances in neuroblastoma: association of miRNA expression with survival. PLoS One. 2009; **4**(11): e7850.
23. Felty Q, Yoo C, Kennedy A. Gene expression profile of endothelial cells exposed to estrogenic environmental compounds: implications to pulmonary vascular lesions. Life Sci. 2010; **86**(25-26): 919-27.
24. Oh JH, Craft J, Al Lozi R, Vaidya M, Meng Y, Deasy JO, et al. A Bayesian network approach for modeling local failure in lung cancer. Phys Med Biol. 2011; **56**(6): 1635-51.
25. Alterovitz G, Tuthill C, Rios I, Modelska K, Sonis S. Personalized medicine for mucositis: Bayesian networks identify unique gene clusters which predict the response to gamma-D-glutamyl-L-tryptophan (SCV-07) for the attenuation of chemoradiation-induced oral mucositis. Oral Oncol. 2011; **47**(10): 951-5.
26. Chou HL, Yao CT, Su SL, Lee CY, Hu KY, Terng HJ, et al. Gene expression profiling of breast cancer survivability by pooled cDNA microarray analysis using logistic regression, artificial neural networks and decision trees. BMC Bioinformatics. 2013; **14**(1): 100.
27. Vineetha S, Chandra Shekara Bhat C, Idicula SM. MicroRNA-mRNA interaction network using TSK-type recurrent neural fuzzy network. Gene. 2013; **515**(2): 385-90.
28. Statnikov A, Wang L, Aliferis CF. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics. 2008; **9**: 319.
29. Guo Y, Graber A, McBurney RN, Balasubramanian R. Sample size and statistical power considerations in high-dimensionality data settings: a comparative study of classification algorithms. BMC Bioinformatics. 2010; **11**: 447.
30. Pirooznia M, Seifuddin F, Judy J, Mahon PB, Potash JB, Zandi PP. Data mining approaches for genome-wide association of mood disorders. Psychiatr Genet. 2012; **22**(2): 55-61.
31. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning New York: Springer; 2009.
32. Sorlie T, Tibshirani R, Parker J, Hastie T, Marron JS, Nobel A, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. Proc Natl Acad Sci U S A. 2003; **100**(14): 8418-23.
33. Dettmer M, Vogetseder A, Durso MB, Moch H, Komminoth P, Perren A, et al. MicroRNA expression array identifies novel diagnostic markers for conventional and oncocytic follicular thyroid carcinomas. J Clin Endocrinol Metab. 2013; **98**(1): E1-7.
34. Uhlmann ME, Georgieva M, Sill M, Linnemann U, Berger MR. Prognostic value of tumor progression-related gene expression in colorectal cancer patients. J Cancer Res Clin Oncol. 2012; **138**(10): 1631-40.
35. Al-Shboul B, Myaeng S. Initializing K-Means using Genetic Algorithms. World Academy of Science, Engineering and Technology 2009; **54**.
36. Pacula M. [cited; Available from: http://blog.mpacula.com/2011/04/27/k-means-clustering-example-python/
37. Iam-On N, Boongoen T. A new locally weighted K-means for cancer-aided microarray data analysis. J Med Syst. 2012; **36 Suppl 1**: S43-9.

38. Eo HS, Heo JY, Choi Y, Hwang Y, Choi HS. A pathway-based classification of breast cancer integrating data on differentially expressed genes, copy number variations and microRNA target genes. Mol Cells. 2012; **34**(4): 393-8.
39. Shi T, Horvath S. Unsupervised Learning With Random Forest Predictors. Journal of Computational and Graphical Statistics. 2006; **15**(1): 118-38.
40. Shi T, Seligson D, Belldegrun AS, Palotie A, Horvath S. Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. Mod Pathol. 2005; **18**(4): 547-57.
41. Hosseinzadeh F, Ebrahimi M, Goliaei B, Shamabadi N. Classification of lung cancer tumors based on structural and physicochemical properties of proteins by bioinformatics models. PLoS One. 2012; **7**(7): e40017.
42. Liaw A, Wiener M. Classification and Regression by randomForest. R News. 2002; **3**(2).
43. Pham DT, Ruz GA. Unsupervised training of Bayesian networks for data clustering. Proceedings of the Royal Society A. 2009; **465**: 2927–48.
44. Agius P, Ying Y, Campbell C. Bayesian unsupervised learning with multiple data types. Stat Appl Genet Mol Biol. 2009; **8**: Article27.
45. Xulvi-Brunet R, Li H. Co-expression networks: graph properties and topological comparisons. Bioinformatics. 2010; **26**(2): 205-14.
46. Xiang Y, Zhang CQ, Huang K. Predicting glioblastoma prognosis networks using weighted gene co-expression network analysis on TCGA data. BMC Bioinformatics. 2012; **13 Suppl 2**: S12.
47. Staehler CF, Keller A, Leidinger P, Backes C, Chandran A, Wischhusen J, et al. Whole miRNome-wide differential co-expression of microRNAs. Genomics Proteomics Bioinformatics. 2012; **10**(5): 285-94.
48. Shen R, Olshen AB, Ladanyi M. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics. 2009; **25**(22): 2906-12.
49. Curtis C, Shah SP, Chin SF, Turashvili G, Rueda OM, Dunning MJ, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature. 2012; **486**(7403): 346-52.
50. Monti S, Tamayo P, Mesirov J, Golub T. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Machine Learning. 2003; **52**: 91–118.
51. Swift S, Tucker A, Vinciotti V, Martin N, Orengo C, Liu X, et al. Consensus clustering and functional interpretation of gene-expression data. Genome Biol. 2004; **5**(11): R94.
52. Hayes DN, Monti S, Parmigiani G, Gilks CB, Naoki K, Bhattacharjee A, et al. Gene expression profiling reveals reproducible human lung adenocarcinoma subtypes in multiple independent patient cohorts. J Clin Oncol. 2006; **24**(31): 5079-90.
53. Freyhult E, Landfors M, Onskog J, Hvidsten TR, Ryden P. Challenges in microarray class discovery: a comprehensive examination of normalization, gene selection and clustering. BMC Bioinformatics. 2010; **11**: 503.
54. de Souto MC, Costa IG, de Araujo DS, Ludermir TB, Schliep A. Clustering cancer gene expression data: a comparative study. BMC Bioinformatics. 2008; **9**: 497.
55. Figueiredo MAT, Jain AK. Unsupervised Learning if Finite Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; **24**(3).
56. Datta S. Evaluation of clustering algorithms for gene expression data. BMC Bioinformatics. 2006; **7 Suppl 4**: S17.
57. He M, Wang Y, Li W. PPI finder: a mining tool for human protein-protein interactions. PLoS One. 2009; **4**(2): e4554.
58. Aggarwal C. Data Streams, Models and Algorithms: Springer; 2007.

59. Samarati P, Sweeney L. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression. Technical Report; 1998.
60. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramanian M. ℓ-diversity: Privacy beyond kappa-anonymity. 22nd International Conference on Data Engineering; 2006; 2006.
61. Li N, Li T, Venkatasubramanian S. t-Closeness: Privacy beyond k-anonymity and l-diversity. 23rd International Conference on Data Engineering; 2007; 2007.
62. Trajkovsky I. Functional Interpretation of Gene Expression Data: Translating high-throughput DNA microarray data into useful biological knowledge; LAP LAMBERT Academic Publishing (2011).
63. http://www.geneontology.org/. [cited; Available from:
64. http://www.genome.jp/kegg/. [cited; Available from:
65. http://www.reactome.org/ReactomeGWT/entrypoint.html. [cited; Available from:
66. Ebell MH, Siwek J, Weiss BD, Woolf SH, Susman J, Ewigman B, et al. Strength of recommendation taxonomy (SORT): a patient-centered approach to grading evidence in the medical literature. J Am Board Fam Pract. 2004; **17**(1): 59-67.
67. Sarker A, Mollá-aliod D, Paris C. Towards Automatic Grading of Evidence. In Proceedings of the Third International Workshop on Health Document Text Mining and Information Analysis (LOUHI 2011). 2011.
68. Choi S, Ryu B, Yoo S, Choi J. Applying Metasearch Technique to Medical Literature Retrieval for Evidence-Based Medicine. SIGIR Workshop on " entertain me " : Supporting Complex Search Tasks,
July 28, 2011, Beijing, China. 2011.
69. Demner-Fushman D, Few B, Hauser SE, Thoma G. Automatically identifying health outcome information in MEDLINE records. J Am Med Inform Assoc. 2006; **13**(1): 52-60.
70. Treadwell JR, Tregear SJ, Reston JT, Turkelson CM. A system for rating the stability and strength of medical evidence. BMC Med Res Methodol. 2006; **6**: 52.
71. Dehing-Oberije C, De Ruysscher D, Petit S, Van Meerbeeck J, Vandecasteele K, De Neve W, et al. Development, external validation and clinical usefulness of a practical prediction model for radiation-induced dysphagia in lung cancer patients. Radiother Oncol. 2010; **97**(3): 455-61.
72. Haerian K, Varn D, Vaidya S, Ena L, Chase HS, Friedman C. Detection of pharmacovigilance-related adverse events using electronic health records and automated methods. Clin Pharmacol Ther. 2012; **92**(2): 228-34.
73. Trifiro G, Pariente A, Coloma PM, Kors JA, Polimeni G, Miremont-Salame G, et al. Data mining on electronic health record databases for signal detection in pharmacovigilance: which events to monitor? Pharmacoepidemiol Drug Saf. 2009; **18**(12): 1176-84.
74. Chazard E, Ficheur G, Merlin B, Genin M, Preda C, Beuscart R. Detection of adverse drug events detection: data aggregation and data mining. Stud Health Technol Inform. 2009; **148**: 75-84.
75. Dart T, Cui Y, Chatellier G, Degoulet P. Analysis of hospitalised patient flows using data-mining. Stud Health Technol Inform. 2003; **95**: 263-8.
76. Brossette SE, Sprague AP, Hardin JM, Waites KB, Jones WT, Moser SA. Association rules and data mining in hospital infection control and public health surveillance. J Am Med Inform Assoc. 1998; **5**(4): 373-81.
77. Hall MEF. The WEKA Data Mining Software: An Update. SIGKDD Explorations. 2009; **11**(1).

78. Kurt Hornik CB, Achim Zeileis. Open-Source Machine Learning: R Meets Weka. Computational Statistics. 2009: 225-32.