UPPSALA
UNIVERSITET

VisuNet: Visualizing Networks of feature interactions in rule-based classifiers

Stephen Omondi Otieno Anyango

Degree project in bioinformatics, 2016
Examensarbete i bioinformatik 30 hp till masterexamen, 2016
Biology Education Centre and Institute of cell and Molecular Biology, Uppsala University
Supervisor: Jan Komorowski

Abstract

Rule-based classifiers have one major advantage over other classes of supervised learning algorithms: interpretability. They provide a means to read into a model and find how the features co-act in order to come to a classification outcome. This in turn enables the researcher to visualize the feature interactions and evaluate the key features that discern between different decision classes. The rules generated from these algorithms, however, can be very many and their analysis is not trivial. This is where proper visualization techniques enable the researcher to filter out clutter and see only important relationships. In addition, the next natural step for genomic data is to find out relationships between the interacting genes and biological networks is always a good starting place. In this study, we introduce VisuNet, a highly interactive, web-based tool for visualization of feature interactions in rule-based classifiers as well as annotation of genomic data with information on biological networks involved. VisuNet can be used with any rule-based classifiers such as decision trees and Rough-Sets, or any model from which rules can be extracted. The tool is hosted online at http://bioinf.icm.uu.se/~visunet/.

Untangling webs of interactions in classification models

Popular Science Summary

Stephen O. O. Anyango

A huge explosion of genomic data has been witnessed in the field of molecular biology due to low-cost of sequencing data with the emergence of new technologies such as Next Generation sequencing (NGS). This has necessitated the use of novel techniques to analyze this kind of data for disease studies and other kinds of research. Classification methods, which are initially trained using data whose outcome/classes are known, produce a model (or classifier) which is then able to assign an unknown object to a class with a certain level of certainty. These algorithms have found wide application in the field of computational biology and medicine. For example, predicting whether a patient will have breast cancer given their DNA is a typical classification problem. This study developed a tool which focuses on one class of such algorithms: rule-based classifiers.

Rule-based classification uses Boolean logic to ascertain whether an object belongs to one set or another (probability 1 or 0) and assuming a probability measure for the cases that are vague. This makes them simple to understand but more importantly allows them to be easily interpreted by domain experts. One common way of interpreting them is by extracting the rules into IF…THEN statements which can be easily understood by most molecular biology experts. The challenge is that these algorithms may produce a lot of rules and hence reading the textual representation is not always plausible. Also, the interactions between the features in the rules are not clear from text and hence the need for proper visualization.

VisuNet has been developed in this study for this specific purpose: It enables the user to interactively view the feature interactions in the classification model as a network whose nodes are features and whose edges are interactions between these features. The user can additionally provide a mapping file, if the data is genomic in nature, and VisuNet will annotate the network diagram with biologically relevant data. The features are specifically annotated with data from KEGG Metabolic Pathways and Gene Ontology terms. These provide an overview of how the function(s) of the genes represented by the features in the network and how they could possibly be interacting within the internal cell network. This is a very important overview since genes relating to a particular pathway (e.g. a cancer pathway of interest to the study) or genes that share certain ontology terms form a good starting point of further investigation. The tool is hosted online as a web application for general use by the scientific community.

Degree project in Bioinformatics, Master of Science (2 years), 2016
Biology Education Center and Institute of Cell and Molecular Biology
Supervisor: Jan Komorowski

Table of Contents

Abbreviations

AJAX – Asynchronous JavaScript and XML

ANN – Artificial Neural Network

CART – Classification and Regression Trees

CSS – Cascading Stylesheets

GIS – Geographical Information System

GO – Gene Ontology

JSON – JavaScript Object Notation

KEGG – Kyoto Encyclopedia of Genes and Genomes

HTML – Hyper-Text Mark-up Language

RNA – Ribonucleic Acid

MCFS – Monte Carlo Feature Selection

MCFS-ID – Monte Carlo Feature Selection and Interdependency Discovery

RF – Random Forests ™

SVG – Scalable Vector Graphics

SVM – Support Vector Machine

UI – User Interface

1 Introduction

Data visualization has become a critical part of analysis and by extension research. A good visualization tool is able to draw the attention of the researcher to critical details not easily visible or clear in the numerous amount of textual data normally output in the process. For instance, it is easy to visualize patterns of high expression in a graphed microarray output presented as a heat map rather than as a matrix of numbers (gene expression levels). Similarly for machine learning algorithms, it is the norm rather than the exception to have a clustering algorithm output the result in a visual plot in addition to the textual data. This speeds up the process of analysis but making subtle clues pop up. A good visualization tool should, in addition to showing the diagram, include a level of interactivity to allow the user explore what they are seeing to some level of detail. For biologists and other domain experts, this ability to visualize your data and interact with it markedly cuts down the time spend in analysis of results. For presentation purposes, the need for good visualization tools cannot be stressed enough.

It is on this premise that this study presents a web-based and highly interactive tool for visualization of networks for rule-based classifier models. With a focus on feature interactions and annotation of genic information with data of biological networks, VisuNet provides a platform for discovery of key drivers for the classification. VisuNet is available online at http://bioinf.icm.uu.se/~visunet for public access.

Following this introduction into the problem, the remaining sections will flow as follows: Section 2 will review some literature and provide a background on machine learning, Section 3 will cover a definition of terms and key formulae used in the application and the report, describe the overall architecture of the tool and how performance was assured, Section 4 will evaluate the key elements of the features of the software and a basic introduction to its working as well as a validation of the tool in comparison to two different studies, Section 5 will be a discussion of possible applications of the tool and a caveat on feature selection followed by a conclusion in Section 6. Finally, Section 7 will preempt some future work.

2 Background: Visualization in Machine Learning

Machine learning is the field of artificial intelligence in which a computer is programmed to learn. Machine learning has grown over recent years in algorithm design and techniques for data pre-processing and visualization, due to an influx of data in many fields including astronomy, biology, and social media among others. From datasets of few attributes in the 70s to petabytes of data in the recent past, the influx has led to many researchers scraping through to identify patterns and automate tasks that would not be possibly done by humans. Generally, there are two areas of machine learning: supervised and unsupervised learning. There exist other paradigms such as reinforcement learning [1] and semi-supervised learning [2,3] although this document will not go into any further detail of these.

Over the past 10 years, there has been a great push in machine learning hand-in-hand with the data explosion. In the field of biology, the sequencing of the first human genome and the massive decline in the cost of sequencing have both contributed majorly to the genomic data explosion. Molecular biologists have been overwhelmed by this surge of data and the field of bioinformatics has gained a new leash of life as they work with the biologists to dig deep into the overwhelming data to discover elusive patterns. Machine learning gives hope that the vast amount of unlabeled data can be grouped/clustered into some functionally or structurally similar cohorts (unsupervised learning), or domain experts can continue with the work of labelling as machines learn and further classify unknown objects off of the experts' work (supervised learning).

2.1 Unsupervised Learning

If the objects being studied have no predefined or known labels, the problem is a case for unsupervised learning. The aim of the machine learning algorithm is thus to identify in the object set, distinct clusters of objects based on some similarity measure of features or attributes. Clustering is the most common example of unsupervised learning. Clustering algorithms can primarily be categorized into two groups: Partition methods and tree-type methods.

Partition methods create a family of clusters (partitions) where each object belongs to just a single partition [4]. To generate such partitions the ideal requirement is that distances between pairs of objects belonging to the same cluster are smaller than distances between pairs of objects in different clusters although this is not usually possible all the time. The *k*-

means clustering algorithm is the most common partitioning method. The algorithm calculates distances between objects starting from centroids, the number of which is provided by the user. The number of centroids is the same as the number of clusters to be obtained. By calculating distances between the objects and the centroids, as well as recalculating the centroids as the mean of distance from all its cluster points, the algorithm is able to group objects into the specified number of clusters. Hierarchical clustering is the most common type of tree-type clustering methods. Tree methods build a tree of clusters that includes all the objects and for which any two clusters are either disjoint or one cluster is a superset of the other.

Clustering has found numerous applications in the field of bioinformatics with clustering of gene expression data being an almost *de facto* example in which genes with similar expression levels are clustered together. Applications in the field of biology have been many, for example: clustering of metagenomics shotgun sequences [5] and clustering of lipid biomarkers in lipidomics [6] among many others. This document will not focus more on unsupervised learning techniques other than this brief introduction as the tool applies only to supervised learning algorithms.

2.2 Supervised Learning

Although unsupervised learning is very useful and possibly the only way of handling unlabeled data, there are enough scenarios in which there exists some information on the labels. This information can be sufficient to train a classifier from. This type of machine learning, in which the algorithm is first trained with data of known discrete outcomes and tested on how well it can classify previously unseen objects in the testing phase, is termed supervised learning, also commonly referred to as classification.

The procedure for classification begins with acquisition of the dataset. A general outline is shown in Figure 1. For genomic datasets, this could be thousands of attributes (features). A domain expert can assist in the selection of requisite fields to reduce the dimension of the dataset [7]. Alternatively, an appropriate feature selection algorithm can be employed to reduce dimensionality. A mix of the two (domain expert after or before an automated algorithm) is also common. Section 2.3 will cover the issues with high-dimensional datasets in slightly more details. After the features to be used have been selected, depending on the algorithm, the data may optionally be discretized or passed as-is into a classifier.

Discretization helps to prevent overfitting of the data which tends to reduce classifier accuracy considerably. A review [7] details the most common steps involved in the pre-processing of the dataset including ways of handling missing data. The data at this stage is a set of features and a label for each object (also referred to as a decision class or outcome). The set of outcomes is mostly a discrete finite set and it is common in the biological field to have binary outcomes e.g. Infected or not, Susceptible to Cancer or not, spliced or not etc. It is also possible to have labels of few options that are not necessarily binary e.g. type of cancer under study (Breast, Lung, and Pancreas). The data is then fed into the algorithm to produce a classifier (or model). The data fed in is called the training data.

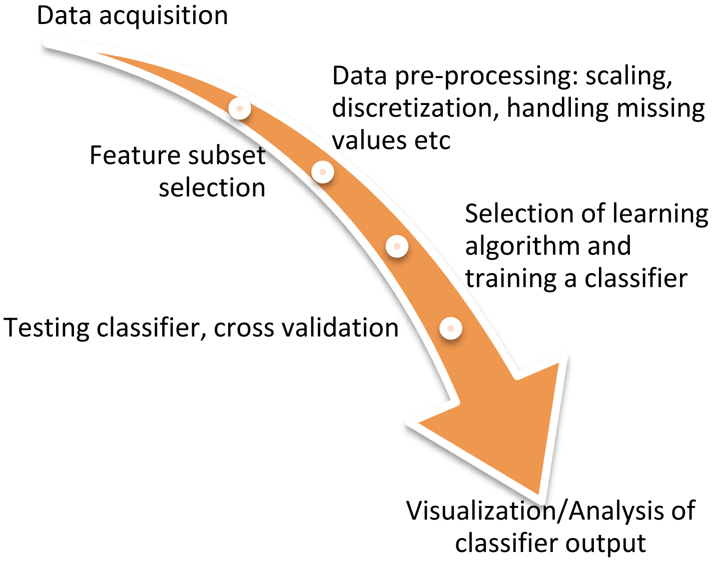

Data acquisition
Data pre-processing: scaling, discretization, handling missing values etc
Feature subset selection
Selection of learning algorithm and training a classifier
Testing classifier, cross validation
Visualization/Analysis of classifier output

Figure 1: A common classification process flow.

The applications of supervised learning in the field of molecular biology are numerous, from classification of novel coding genes and small RNA, to differential gene expression studies in genome-wide association studies, among many others. In this field, it is important for researchers to explicitly consider the aim of the classification tasks: Performance versus interpretation. In many other areas of application like business or even medicine, predictive power seems like a logical choice over interpretation. For instance, a model should be able to correctly diagnose a patient other than wrongly do so and provide an explanation on how it arrived at the decision. Nevertheless, in research, it is prudent that the algorithm be readable. For instance, if a classifier can identifier an object with 99.9% accuracy as susceptible to breast cancer, it is indeed a very desirable classifier; if it can go into the details of how it arrived at the conclusion – for example: “if Gene1 is upregulated and Gene6 downregulated and Gene 8 unchanged then breast cancer susceptibility” – then the research has a more precise target(s) to work with. For the purposes of this document, we consider two groupings of supervised learning algorithms: Black-box vs White-box algorithms.

2.2.1 Black-box classification algorithms

Black-box classification methods are able to produce a model which can classify unknown objects but their working is usually hidden in the algorithm implementation and hence one cannot extract the mechanism by which they decide on the class of an object. They tend to outperform white-box in terms of accuracy but are not quite feasible for most biological research scenarios where the researcher would like to read into the model to know what are the features, for example genes, that are responsible for specific outcomes and how they interact in making up the decision. This paper focuses on classification methods that can produce such rules that make up the classifier rather than black-box methods.

Nevertheless, there has also been considerable effort in research to unravel these black-boxes. In this section we will cover three examples of these algorithms that are common: Support Vector Machines, Artificial Neural Networks and Random forests.

2.2.1.1 Support Vector Machines (SVMs)

SVMs have gained widespread fame because of their high performance as multi-class classification algorithms. Although initially used for binary classification [8], they have been adapted for use in multi-class problems and have recorded very good performance and hence been adopted widely in the biological field. They have been applied in functional annotation of fungal genomes [9], longitudinal studies [10] just to name a few.

The SVM algorithm uses kernel functions to project data sets into higher-dimensional space representations in which a linear separation of positive and negative training instances is feasible [11]. The major challenge with SVMs has been their perception as black-boxes for which no explanation on why classification fails or succeeds. Nevertheless, there have been attempts to modify the kernel to allow interpretations [11–15].

2.2.1.2 Artificial Neural Networks (ANNs)

Created to mimic biological neural network, ANNs provide a fast and well-performing algorithm for classification. Neural networks consist of an input layer of neurons or nodes, one or more hidden layers and an output layer of neurons [16]. Neural networks have found much application in numerous fields of computational biology [4,17]. A list of applications and potential areas are covered by [18,19]. Like SVMs, they are largely considered black-boxes and there is much effort [20,21] to look inside the hidden layers and possibly extract information that can be used to unveil the workings of the output models

A rising category of neural networks are deep learning algorithms [22] that model hierarchical abstractions in input data with the help of multiple layers. They can have a huge parameter space and therefore can be compute intensive [23]. In addition to computer vision, speech recognition and natural language processing, they have also found application in the fields of genomics [22,24,25] and drug-discovery and continue to show great promise even though they still face the challenge of interpretation.

2.2.1.3 Random Forests (RFs)

According to Breiman (2001), RFs are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. RFs extend the concept of decision trees (described in Section 2.2.2.1) but unlike the latter, RFs do not produce an explicit model hence are in this category of black-boxes [26]. They also provide - as part of the process - a ranking of the features even though this is not necessarily used by the algorithm itself [27].

RFs have also been widely used in the field of genomics including gene classification [28] and various genome-wide association studies [29,30].

2.2.2 White-box classification algorithms

White-box algorithms have one main characteristic in common: interpretability. One common group of these algorithms is ones that produce explicit rules generally named rule-based classifiers. Rule-based classification algorithms such as Rough Sets [31,32] and Decision Trees can be easily translated into an IF-THEN model; The output is then easier to visualize for domain experts to make deductions from the model. Their level of complexity is not as high as most black-box methods; decision trees may perhaps be the simplest models since they could even be constructed by hand for very simple data. These models also perform comparably with black-box methods in accuracy and speeds depending on algorithm implementation and dataset in question.

2.2.2.1 Decision Trees

A decision tree classifies data items by posing a series of questions about the features associated with the items. A simplified representation of a classification tree is shown in Figure 2. Each question is contained in a node, and every internal node points to one child node for each possible answer to its question [33]. An optimal decision tree attempts to reduce the depth (number of questions) [34] without compromising classification accuracy. This increases legibility as a deep tree increases complexity.

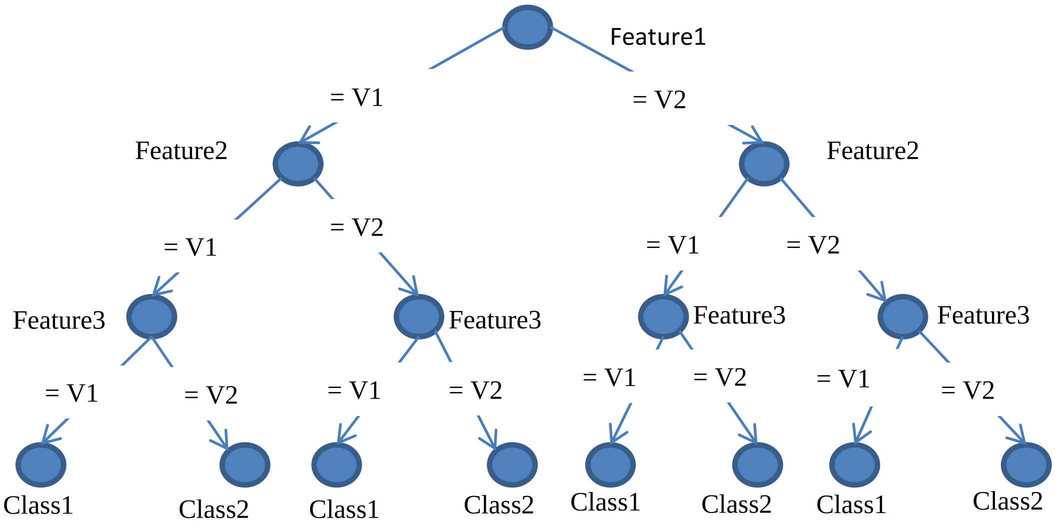


Figure 2: Decision Tree Layout. Each non-leaf level represents a single feature with the edge being a value that the feature takes. Leaf nodes show the outcome of classification while the root node does not represent a feature.

Applications of decision trees have been plentiful in the fields of computational biology [35–37] and medicine [38–40]. Rules can easily be extracted from these trees, a process named discrimination [34]. Implementations of trees have been done in many algorithms including MCFS (a filter for feature selection that uses classification trees) [41], C4.5 [42] and CART [43].

2.2.2.2 Rough-sets

Introduced by Pawlak (1982), Rough sets are a formal approximation of crisp sets. Built on Boolean reasoning, they aim to find a minimal combination of features, called Reducts, that discern between classes [44]. This makes them ideal for feature selection as well as classification. It is also possible to extract rules from the Reducts using various algorithms. ROSETTA [32] is an implementation of Rough sets that is able to perform the pipeline for classification in an easy to use Graphical User Interface (GUI) or a feature-rich command-line interface. The output from the training is a set of rules with quality measures (support, accuracy, coverage and strength) that enable a ranking of rules [31]. As output rules of ROSETTA can be numerous depending on the reduction algorithm used, the ranking of the rules and filtering options enables the user to consider a subset of the rules that sufficiently cover the dataset. The filtered rules can then be fed into VisuNet for visualization.

Rough sets have also found much application in the field of computational biology [45–48] and perform quite comparably with other black-box methods depending on dataset complexity. The legible model produced allows the researchers, who would like to know the contribution and interaction of features do particular decision, to decipher the model and investigate the features further.

2.3 Feature selection

Most datasets in the field of biology tend to have features far greater than the number of objects. It is common to have a hundred or even fewer patients while observing tens of thousands of genes in gene expression data. Such high-dimensional cases are commonly referred to as the “small n large p problems” and present several challenges to the classification algorithms. This has been rightly named the ‘curse of dimensionality’ [49]. First of all, rarely for such problems do many of the features have requisite predictive power; they are either irrelevant or redundant. In terms of running time, the numerous features greatly impact the running time, especially since some algorithms scale very poorly with higher dimensions. Moreover, the high dimensionality will also lead to over-fitting – describing a random error or noise rather than an underlying data – thereby reducing classification accuracy. This is where feature selection (FS) or sub-setting algorithms come in; they select the set of features that has the most discriminative information from the original feature set.

However, in implementation of FS algorithms, a single feature may be considered irrelevant based on its correlation with the class, but it may become very relevant if combined with other features. The unintentional removal of these features can result in the loss of useful information and thus may cause poor classification performance [50]. It is therefore important that the FS algorithm takes into account interacting features or correlation between features. This is also a well-studied area and a lot of algorithms have been developed that consider some metric of correlation between features or feature subsets to the decision class [6,51–54]. It is this correlation that the tool described herein builds hopes to unveil by visualization.

The uses of feature selection (FS) techniques are many fold. FS avoids the over fitting problem thereby improving the performance of classification models, develops fast and cost effective models, facilitates data visualization, reduces the measurement and storage requirements, reduces training and testing time of the prediction model [52] and enhances comprehensibility of learned results [51]. There are mainly three types of FS methods: wrappers that are wrapped around a classifier, filters that work as a pre-classification step, and embedded methods that are part of the classification algorithm. Embedded and wrapper methods are therefore tightly coupled to a classifier. The filter model evaluates the goodness of feature with pre-specified criteria, which is independent of learning algorithms [55].

In this study, we validate VisuNet against data studied by [56] and used to validate MCFS-ID, described later in Section 2.4.2. MCFS-ID is based on the Monte-Carlo Feature Selection (MCFS) algorithm [57]. The MCFS algorithm is a filter algorithm that selects a feature if it is likely to take part in the process of classifying samples into classes ‘more often than not’. It employs classification trees to calculate the relative importance (‘readiness’ of a feature to take part in the classification process) of a feature. The algorithm then ranks the features on basis of relative importance (RI) and provides a statistically-advised cut-off point. Features above the cut-off point are deemed sufficient to build a good classifier. We comment on similarities and differences between the results arrived at by the original paper, MCFS-ID and VisuNet.

2.4 Visualization in Networks

2.4.1 Why Networks

Networks provide an intuitive and natural way to interpret interactions and relationships. A lot of studies began with exploration of networks in nature and networks have been extrapolated into various inventions and algorithm development by human beings. For instance, networks occur in man-made transport networks – rail, flight, roads – that have now been mapped into GIS software and are easy to find information on, using various map-providing software and websites like Google Maps. Also, the internet and social media networks that have almost become indispensable to human life in many regions of the world [58]. Even natural biological networks (metabolic reactions, neural networks, blood circulation, food webs) [59] have been and are currently being explored in various facets of science to discover the functioning of cells at a large scale. This is what makes network layout of data so intuitive. In this report, the terms graphs and networks may be used interchangeably.

A formal definition of graph is given as follows according to [60]:

> A graph is an ordered triple $G = (V(G), E(G), I_G)$, where $V(G)$ is a nonempty set whose elements are vertices (or nodes or points), $E(G)$ is a set whose elements are edges, disjoint from $V(G)$ and I_G is an "incidence" relation that associates with each element of $E(G)$ an unordered pair of elements (same or distinct) of $V(G)$; $V(G)$ and $E(G)$ are the vertex set and edge set of *G*, respectively. If, for the edge *e* of G, $I_G(e) = \{u, v\}$ we write it as: $I_G(e) = uv$.

For example, given the sets:

$$\mathrm{V(G)} = \{v_1, v_2, v_3, v_4\}$$

$$\mathrm{E(G)} = \{\mathrm{e}_1, e_2, e_3\}$$

$\mathrm{I_G(e_1)} = \{v_2, v_4\}, \mathrm{I_G(e_2)} = \{v_1, v_4\},\ \mathrm{I_G(e_3)} = \{v_2, v_3\}$

In this case, $G = (\mathrm{V(G), E(G), I_G})$ is a graph. A simple graph is a graph with neither loops - $\mathrm{I_G}\left(\mathrm{e_j}\right) = \{\mathrm{v_i, v_i}\}$ - nor parallel edges (edges with the same start and end vertices). A complex graph can contain loops and multiple/parallel edges. An ordered/directed graph is one in which the direction of the edges are taken into account such that if $\mathrm{I_G}\left(\mathrm{e_j}\right) = \{\mathrm{v_i, v_{i+1}}\}$ and

$\mathrm{I_G}(\mathrm{e_{j+1}}) = \{\mathrm{v_{i+1}}, \mathrm{v_i}\}$, then $\mathrm{I_G}(\mathrm{e_j}) \neq \mathrm{I_G}(\mathrm{e_{j+1}})$. In an unordered or undirected graph, $\mathrm{I_G}(\mathrm{e_j}) = \mathrm{I_G}(\mathrm{e_{j+1}})$. A graph is complete if each node has at least an edge connecting it to another, and incomplete if there is at least one node that has no edge. This tool presented in this paper will only produce simple, unordered and sometimes incomplete graphs. This is because the determinant of directionality would imply causality or some kind of flow, but we are more interested in the correlation. There is a possibility for causality but this could be merely as an artifact of the data rather than a rule.

Graphs are easy to interpret into diagrams and hence the basis for this work. The above graph can be represented as shown in Figure 3. We seek to represent the rules input into the tool as a network of interactions between interacting features in the rules and allow for annotation of nodes and edges with biological networks for genomic datasets.

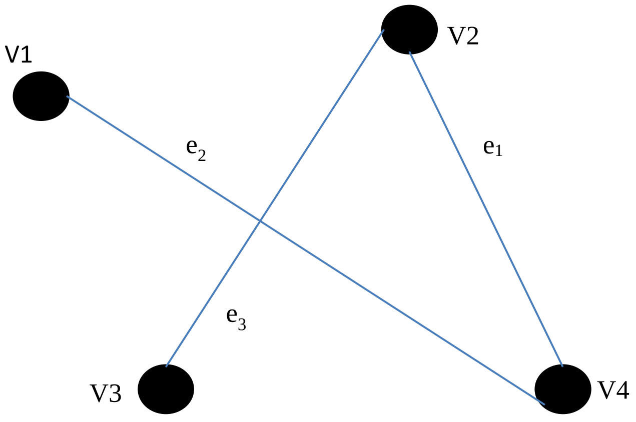
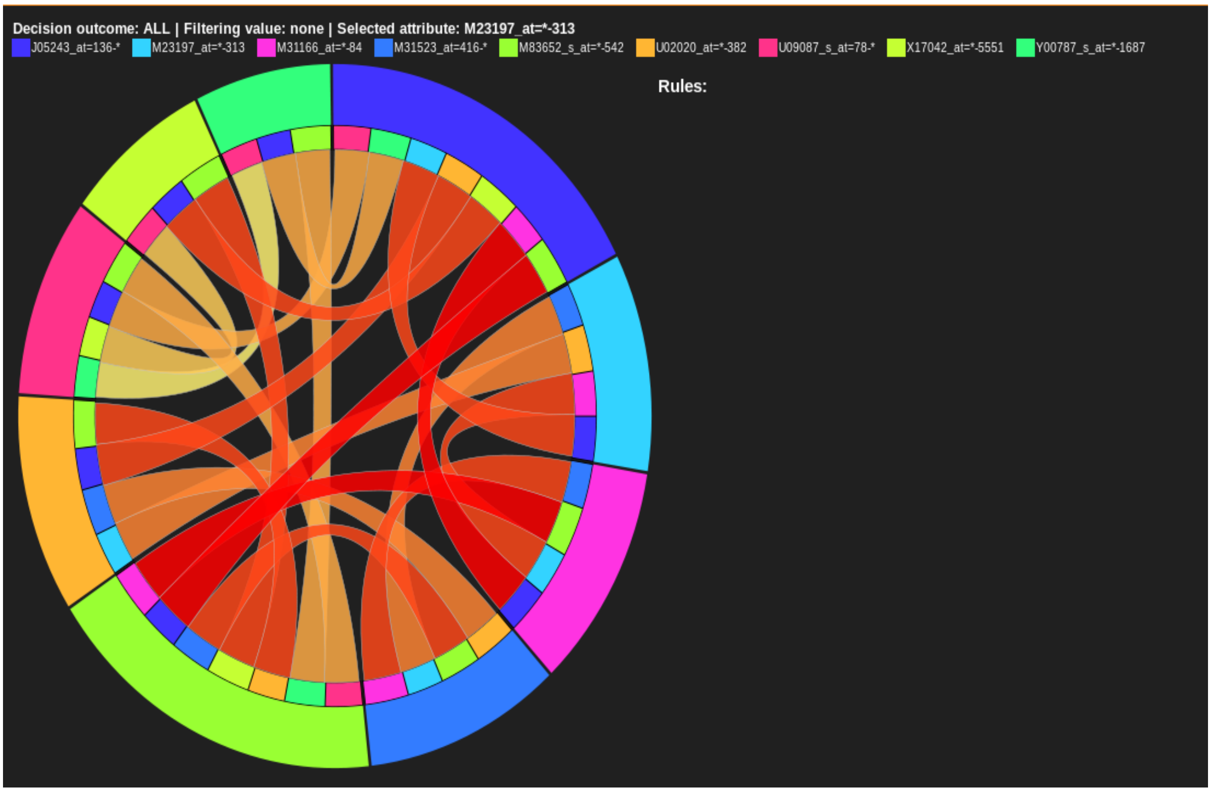


Figure 3: Diagrammatic representation of a simple, unordered, complete graph. Graph has 4 vertices (v1,v2,v3,v4) and 3 edges (e1,e2,e3).

2.4.2 Similar Works

Some tools have already been developed for the visualization of rules. Some are generic like Ciruvis [61] while others are tied to specific classifiers, for example MCFS-ID [54] and, Mosaic Plots [62] and arulesvis [63] R package that are specific to association rules to name a few. Ciruvis represents interactions between features in rule based classifiers in a closed circular form. The user can provide a grouping and coloring scheme in separate files for a customized view. Since it uses Scalable Vector Graphics (SVG) for its output, it provides a basic level of interactivity for showing labels, highlighting interactions on hover, and showing rules that fire for a highlighted interaction. It provides the interactions and weighted view of features per decision class and an overall view for all decisions. The tool is intuitive and aesthetic producing good quality production ready images. A sample output of Ciruvis is shown in Figure 4.

The Monte-Carlo Feature Selection and Interdependency Discovery (MCFS-ID) tool borrows part of its name from the feature selection algorithm, MCFS, upon which it is built. MCFS-ID uses a visualization of interdependencies between the selected features in a network layout. It colors nodes based on their MCFS calculated RI of the features assigning the strongest intensity to the most important feature and reducing the intensity with reduced RI. The graph produced is a simple, ordered and incomplete graph having arrows in the edges pointing to the node with better RI. The thickness of the edges is the weighted strength of the interaction while the size of nodes grows with the number of edges going into it.



Decision outcome: ALL | Filtering value: none | Selected attribute: M23197_at=*-313

Figure 4: A sample view of Feature Interactions in Ciruvis. Rule networks for Acute Lymphoblastic Leukemia related genes as described by [61].

Nevertheless, MCFS-ID does not provide much interactivity and is tightly coupled with the MCFS method for feature selection. It does not regard the feature ranking per decision class and hence cannot provide discerning features between the classes. Despite these few limitations, it provides an intuitive view of the feature interdependencies and provides a good starting point for analysis.

There exist other tools that have been employed in association rules used in data mining. The *arulesviz* [63] package in R provides a way of visualizing association rules in several formats. It allows visualization of rules in graph format – shown in Figure 5 - or even hierarchical grouping [64] displayed as a matrix. It is designed for use with the *arules* [65] package in R and hence is tightly coupled too.

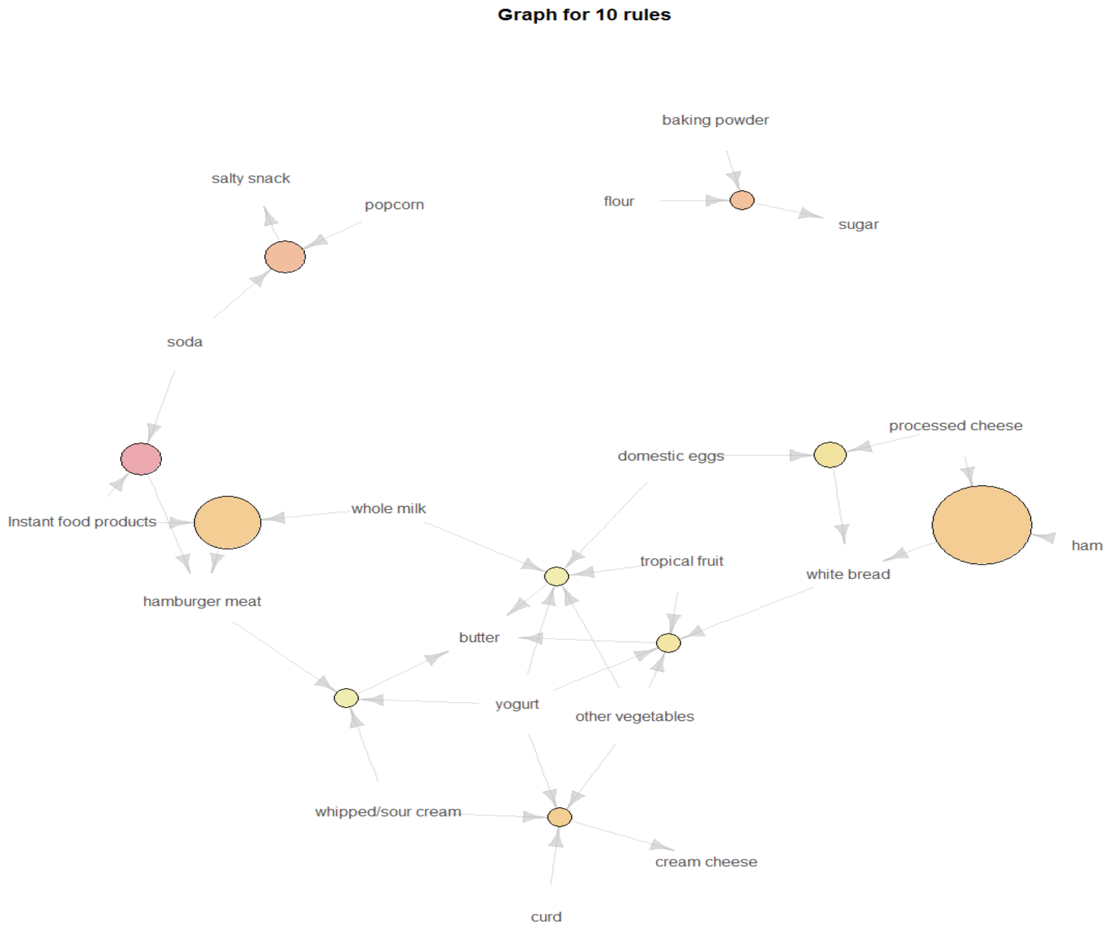


Figure 5: Graph visualization of 10 rules from the Groceries dataset. Visualization was done using the R package *arulesviz* according to instructions in [64].

Graph-based visualization offers a very clear representation of rules but they tend to easily become cluttered and thus are only viable for very small sets of rules. This is the challenge with most non-interactive graph tools. Ability to zoom, selected subsets of data, search and filter and hence very necessary when visualizing large set of rules. A key thing in graph-based visualizations is also to provide labels that are of interest to the user. Usually, graph visualizations annotate nodes and edges with labels and adjust node coloring and edge widths to improve visual effect and highlight key nodes and edges.

2.5 Aim of study

Domain experts are usually not also power users and require user interfaces that are intuitive and informative so as to avoid an overhead of learning the complex tools. Also, interactive data visualization allows the user to control clutter and focus on areas of interest quickly. In this study, we present VisuNet: an interactive, web-based visualization of feature interactions in form of a simple, labelled, unordered and sometimes incomplete graph. The tool should allow as input rules formatted in a specified format or from ROSETTA, and provide a searchable and filterable view of the feature interactions in the input rules per decision class with ability to zoom. We hypothesize that annotation of such interactions with biological networks for genomic input will not only cut the time spent foraging through multiple genomic databases for information but also unravel interesting relationships at a glance. We employ Gene Ontology (GO) terms and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways for the initial version of the tool.

3 Method

3.1 Definition of terminology

For consistency, the previous definitions defined by [61] were kept with a few changes to incorporate the differences in visualization. A rule has the form:

IF *feature=value [, feature=value]* THEN class=decision

Where *feature='value'* (for example MIF='high') is referred to as a condition. Henceforth, the words condition and feature may be used interchangeably. Conventionally, the IF part is called the antecedent or simply left hand side (LHS) and the 'THEN' part is termed consequent, or the right hand-side (RHS). A rule may have one or more conditions in the antecedent. Each condition is represented in VisuNet as a red-colored node. The stronger the color intensity of the condition, the higher the quality of the feature, that is, the feature occurs in top ranked rules. The ranking of a condition x is based on connection defined as:

$$connection(x) = \sum_{r \in R(x)} support(r) \cdot accuracy(r)$$

Where $R(x)$ is the set of all rules with condition$x$. Similarly, two conditions are connected vertices/edges of a graph if they co-occur in some rule(s) and the score of the connection – which determines the thickness of the edge - between two conditions x, and y is defined as:

$$connection(x, y) = \sum_{r \in R(x,y)} support(r) \cdot accuracy(r)$$

Where $R(x, y)$ is the set of all rules in which x and y co-occur. A higher value of connection between any two nodes implies a thicker edge between them.

The size of each node is determined by the number of edges emanating from it. It is important to note that a node may have many connections and hence be large although the connections are of low value hence not as stronger in color. The connection of the edges and nodes are weighted and scaled automatically in the visualization to produce a visual experience that will help discern the strength of the interactions.

3.2 Architecture

3.2.1 Presentation Layer

VisuNet is designed as a three-layer web application as shown in Figure 6. The presentation layer allows the user to interact with the application in two main web pages. It is built using HTML5, CSS3 and JavaScript. HTML5's form error handling is leveraged for simple client-side error handling. Other errors are displayed on this layer using AJAX although propagated from the business logic. Three major JavaScript libraries are used in the application: Bootstrap, JQuery and vis.js.

Bootstrap is used for styling of components due to its adaptability to multiple screen sizes, theming and aesthetic effect. JQuery has become a *de facto* library for ease of writing JavaScript code that improves legibility and together with CSS3 selectors, reduces lines of code considerably. It also provides various commonly-used functions not included in plain JavaScript.

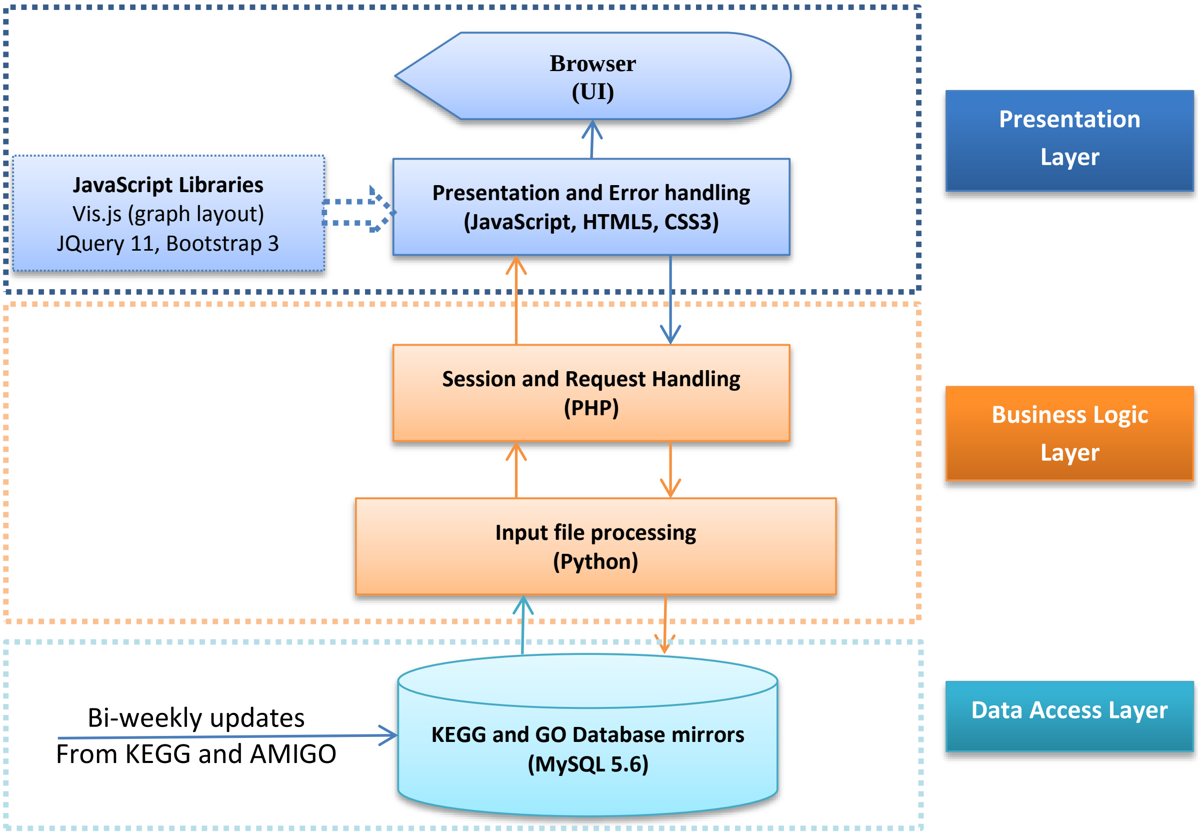


Figure 6: High-level architecture of VisuNet.

The vis.js library is the core of the visualization in the presentation layer. It provides a network module that can create graphs from data in JSON format. It is able to handle large amounts of dynamic data. It uses HTML Canvas object which gives better performance than SVG although at the cost of difficulty obtaining vector-based graphics. It also provides functionality to automatically layout networks using the force-directed placement [66–68] which treats each node as ball and each edge as a spring. By considering repulsion between nodes dependent on their 'mass' property, the nodes avoid overlapping and the spring edges prevent them from moving too far. In a force-directed network layout attraction between nodes is based on their connectedness such those that are connected will attract one another while those repelling those that they are not connected to. The achieved effect is an aesthetically pleasant network view that attempts to minimize overlaps between nodes. This is why force-directed placement is one of the most successful and commonly used automatic graph-layout algorithms and has been implemented by several graphing libraries [69,70] .

3.2.2 Business Logic Layer

To allow ease of access while maintaining a simple-to-maintain codebase, the application was made to be web-based and hosted on a secure web server running Apache 2.4. I also chose to use cross-platform languages to circumvent the need for cross-platform dependencies and hence can easily be hosted on any server platform with minimal changes. The core of the application (Business Logic Layer) is built using PHP 5.5 and Python 2.7.11. This was to leverage Python's scripting novelty while leveraging PHP's ease of session handling and interworking with JavaScript and HTML. The option of using Python's web-enabling libraries, such as Flask and Django, was considered but avoided to reduce the number of external libraries needed to install the application. The web server used to serve pages and process PHP is Apache Web Server 2.4.

In this layer, the user's uploaded input files are processed. A local copy is saved during the duration of the processing and deleted once the processing is completed. A JSON containing the nodes, edges and rules is sent back to the presentation layer for display of the graph. Since given *n* nodes, the maximum number of edges – if all nodes are connected – is *n* x *n,* it is important to limit the amount of data going into the client side by doing some pre-filtering on the data. The home page should provide some defaults (e.g. 0.7 for minimum accuracy, 70% for threshold).

Additionally, the nodes and edges (interactions) are ranked by connection per decision. An extra decision, named 'all' contains all the interactions irrespective of the decision. This provides an overview of the major interactions in the dataset independent of their discerning power.

3.2.3 Data Access Layer

KEGG pathway diagrams can be accessed from the KEGG website which has a flexible URL structure. We leverage this URL structure to create links to the website to visualize a gene of interest in a selected pathway. In order to do this, we could query the KEGG data using their REST services but this proved very slow. Instead, we chose to host the necessary data (organisms, their genes and pathways for those genes) locally in a MySQL database. The same was the case with the GO databases. Another challenge was to allow the user to input gene symbols (BRCA1, IFIT2) in the extra data file since several databases use varying ID formats and, the HGNC format is not yet used by KEGG and Go databases. A RESTful API provided by *bioDBnet* [71] is used to convert the KEGG gene ID's into an official gene symbol.

In-house scripts are used to keep the two databases up-to-date; they are run bi-weekly. All the scripts are bundled in the source code available on request.

3.3 Performance

For KEGG and GO terms, a local mirror of relevant subsets of the two databases was made so as to improve performance. In-house automated scripts are available to update the databases periodically. This is not so frequent since the two databases are not real-time-growing; Weekly or even monthly updates would be sufficient. These can be easily made into scheduled jobs run automatically by the operating system. Database tuning was vital for performance of this tool in a bid to reduce processing time. To this effect each query was timed to find an optimal plan. Indexes were created on the necessary fields to decrease the query time. The database does not grow hence little need for maintenance. Each time a data file is uploaded, the mapping is loaded onto the database for querying. The table is created dynamically using a random ID to allow multi-user access. The table is dropped once the data is moved to the client-side for viewing. Similarly all the files uploaded are deleted on completion of the data processing and transformation steps.

For visual performance (loading and interactivity), the library used (vis.js) is light-weight hence loads quite fast. The library also provides a fast filterable abstraction of data. Tests on load times were performed using Google Chrome's and Mozilla Firefox's developer tools. AJAX data fetches that allow some tasks to be performed in parallel in the background also helped gain performance advantages. Nevertheless, there is still room to improve load times for large rule files (over 20MB). Parallelization of the processing stage is in the action plan.

4 Results

4.1 Features

VisuNet allows a user to select two types of input files for rules on the landing page Figure 7: Line by line and ROSETTA. The latter is from the ROSETTA application. The former is a generic four-column tab-separated file containing - for each rule - a comma-separated list of features in the antecedent of the rule, a comma-separated list of decisions, accuracy and support of the rules which can be written as:

```
Cond1[,Cond2…]<tab>Decision1[,Decision2…]<tab>accuracy<tab>support
```

For example, the below rule:

IF Gene1= "high" and Gene2 = "low" THEN "Breast"
(Accuracy: 0.98 Support: 20)

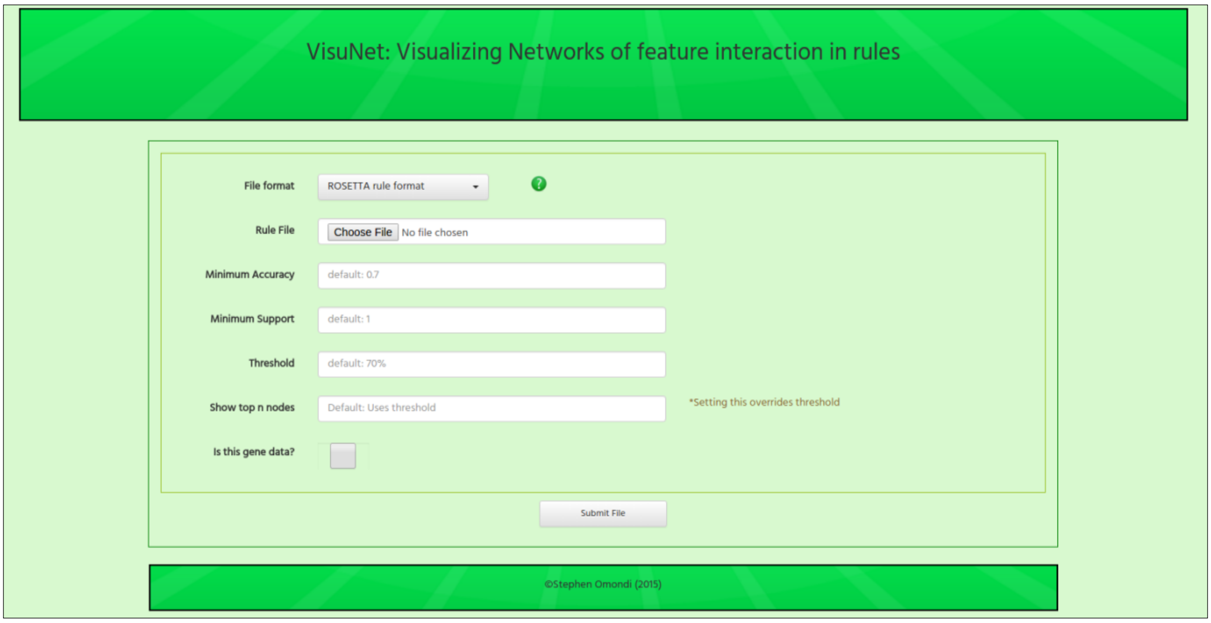
Can be shown in one line in the input data file:

Gene1="high",Gene2="low"<tab>"Breast"<tab>0.9<tab>20

This data file could easily be hacked to have any other weighting method for the rules. The user also is able to select a second two-column tab-separated file mapping each condition to a gene name e.g.

Gene1<tab>BRCA1
Gene2<tab>IL6

Additionally, the user can filter the rules by providing a threshold of the nodes to be shown. The default (70%) shows the top 70% of the nodes/features. This can be reduced to reduce clutter depending on the number of rules.



VisuNet: Visualizing Networks of feature interaction in rules

File format: ROSETTA rule format

Rule File: Choose File No file chosen

Minimum Accuracy: default: 0.7

Minimum Support: default: 1

Threshold: default: 70%

Show top n nodes: Default: Uses threshold *Setting this overrides threshold

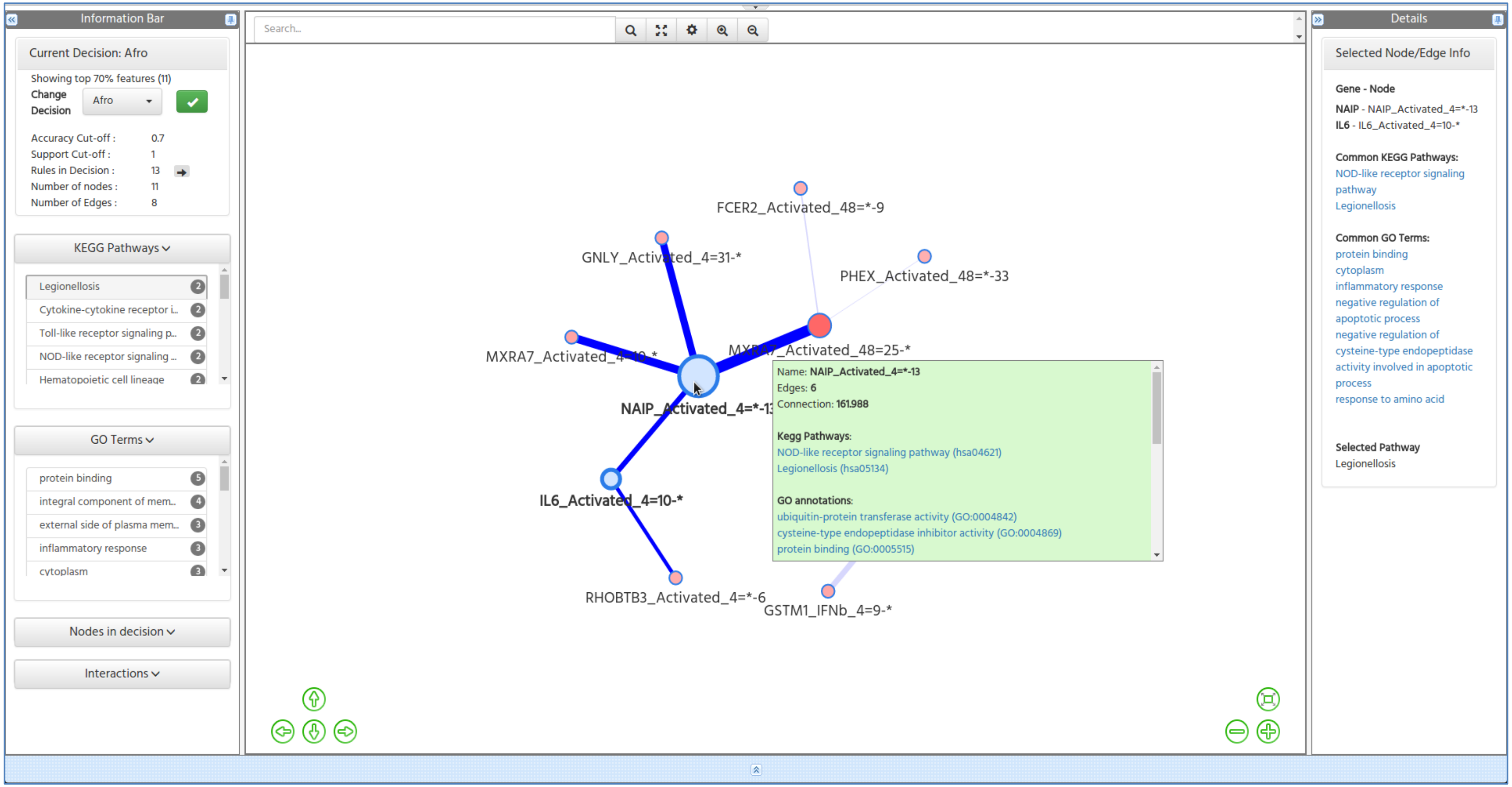
Is this gene data? □

Submit File

©Stephen Omondi (2015)

Figure 7: A screenshot of the VisuNet home page. User can select the rule file, filters and optionally select a data file for genomic data.

After preprocessing, VisuNet presents to the user an interface such as one shown in Figure 8. The user can change decision classes to see the interactions in that particular class. In addition, VisuNet provides search feature to locate nodes, a tabular view of the rules making up the node/edge, a full-screen option enhancing the canvas size in addition to collapsible side panels. The collapsible KEGG and GO term panels can also be used to select nodes; the user selects a pathway/term and all the nodes in the pathway/term are highlighted. The canvas provides a save image option (dependent on browser) that can be used to export to PNG format. The user can also take good quality screenshots in full-screen mode. To even focus further on nodes of interest, the user can filter the view by only showing selected nodes and their interactions. Scrolling the mouse of button zooms in or out increasing visibility and clarity of text. For genomic data for which an input data file has been provided, the KEGG pathways, genes and GO terms link to external databases (KEGG, GeneCards and AMIGO respectively) for more detail.



Current Decision: Afro

Showing top 70% features (11)

Change Decision: Afro ✓

Accuracy Cut-off : 0.7

Support Cut-off : 1

Rules in Decision : 13

Number of nodes : 11

Number of Edges : 8

Legionellosis	2
Cytokine-cytokine receptor i...	2
Toll-like receptor signaling p...	2
NOD-like receptor signaling ...	2
Hematopoietic cell lineage	2

protein binding	5
integral component of mem...	4
external side of plasma mem...	3
inflammatory response	3
cytoplasm	3

Name: **NAIP_Activated_4=*-13**

Edges: **6**

Connection: **161.988**

Kegg Pathways:

NOD-like receptor signaling pathway (hsa04621)

Legionellosis (hsa05134)

GO annotations:

ubiquitin-protein transferase activity (GO:0004842)

cysteine-type endopeptidase inhibitor activity (GO:0004869)

protein binding (GO:0005515)

Selected Node/Edge Info

Gene - Node

NAIP - NAIP_Activated_4=*-13

IL6 - IL6_Activated_4=10-*

Common KEGG Pathways:

NOD-like receptor signaling pathway

Legionellosis

Common GO Terms:

protein binding

cytoplasm

inflammatory response

negative regulation of apoptotic process

negative regulation of cysteine-type endopeptidase activity involved in apoptotic process

response to amino acid

Selected Pathway

Legionellosis

Figure 8: Network visualization in VisuNet. The nodes in the graph represent features while the edges are co-occurrences of the features in rule. The thicker the edges, the more the number of co-occurrences of the two nodes, weighted by the quality of the rules they appear in.

4.2 Validation

To validate the application, two real datasets were employed. The first dataset of rules, taken from [54], was used for biological validation of the MCFS-ID tool. The dataset contains gene expression levels of 236 genes, of which 7 were control, in CD4$^+$ T cells measured after 4 and 48 hours. In the protocol to purify and activate the CD4$^+$ T cells, they were either (1) activated in unbiased conditions (labelled *Gene_Activated_4 or Gene_Activated_48)*, or (2) in biased conditions toward T helper 17 (T$_H$17) (labelled *Gene_Th17_4 or Gene_Th17_48*), or (3) with addition of IFN-β (labelled *Gene_IFNb_4 or Gene_IFNb_48*). The CD4+ T cells were sampled from human blood from 348 healthy patients who were of three different ancestries: European (Caucasian), Asian, and African-American (abbreviated Afro herein). The original aim of the study by [56] had been to investigate variability in immune responses and uncover the genetic drivers for this variation.

Ye *et al* (2014) have reported that the ancestry of the donors markedly influenced the responses with a stronger T$_H$17 associated with the Afro group. In their study, they use a linear model to reveal an ancestry-differentiated expression of 94 out of the 229 genes. They reported, as shown in Figure 9, an overexpression of response genes for donors of African ancestry, lower for European ancestry, and a mixed pattern for Asian ancestry. Also notable was the high expression of the *GSTM1* gene in the Afro group and the *UTS2* gene in the Asian population. In addition, the study noted that the differentially responsive genes include key indicators of TH phenotypes, IL17 family cytokines (over-induced in individuals of African ancestry), and IFNG, which showed an opposite pattern. Over all, there was a notable differential expression of transcripts encoding cytokines, chemokines, or their receptors. For the purposes of brevity, we shall refer to this group as the original study.

A study by [54] also analyzed the same dataset with a focus on ancestry differentiation by the genes under study. They used it to validate MCFS-ID, described previously herein, which graphs interdependencies between features. In the study, they considered the top features generated from MFCSID and mapped their interactions as shown in Figure 11. According to the top features, the first five features represented the UTS2 gene in all their activation states. They further used the ROSETTA application to create a rule-based classifier and visualized its feature interactions using Ciruvis (not shown here; see [54] for figure and links). We shall henceforth refer to this study that used MCFSID as the second study.

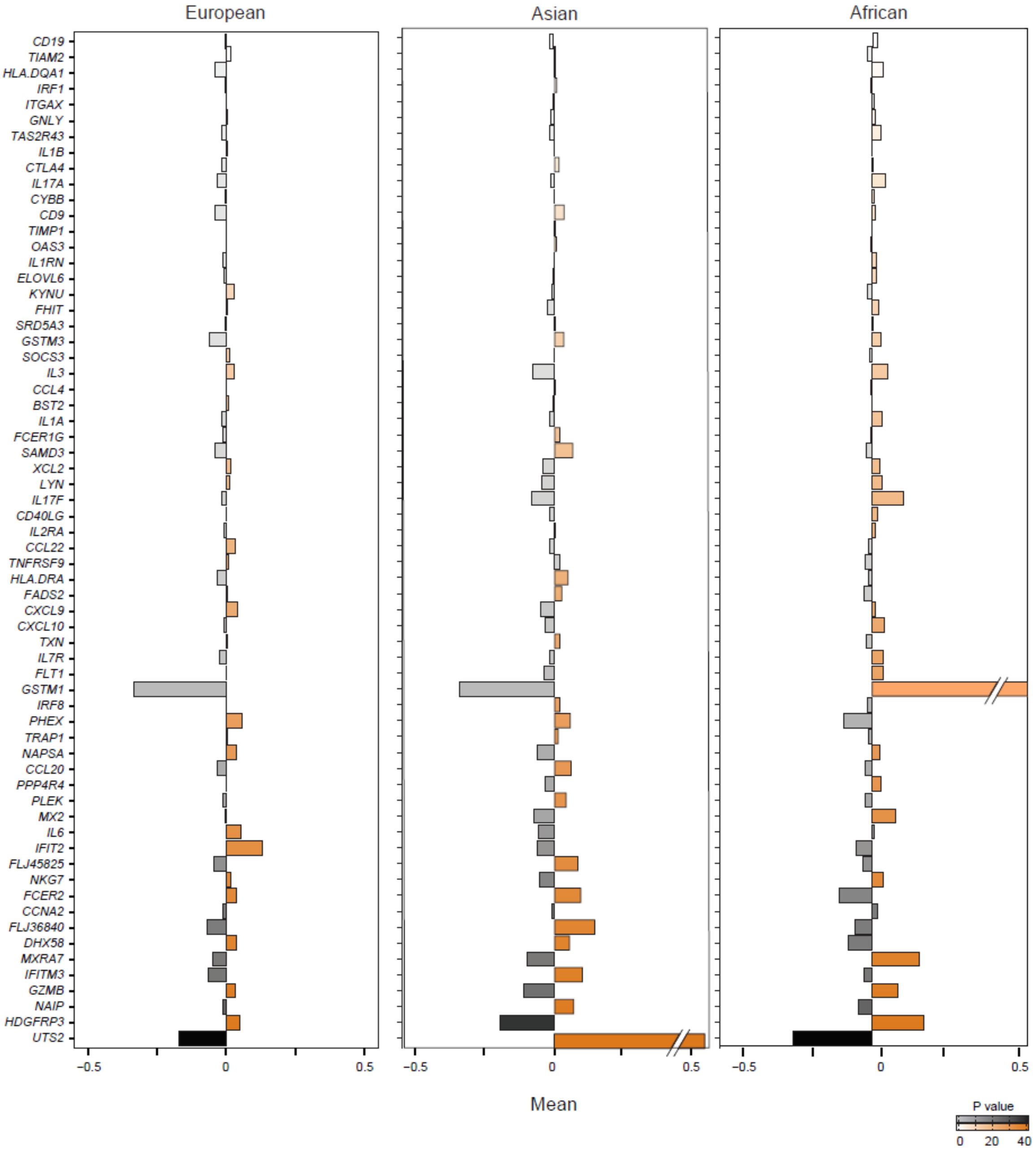


Figure 9: Differential gene expression between the ancestries. Percent difference of average population expression (median) from overall average (median) that shows population differentiation in expression. Retrieved from supplementary materials of [56].

In this study, I have compared the rules generated by second study on ROSETTA using the top 100 features according to MCFS (as a pre-classification feature selection step). I used a minimum accuracy of 0.5 and a default support of 1. The results were visualized using

VisuNet as shown in Figure 10. There were notable similarities and differences from the first and second studies. First the top nodes - *GSTM1* in the African class and *UTS2* in the Asian class - corresponded to the notably high expression levels of these two genes observed in Figure 9 by the first study. The general overexpression in the African population is also notable from the list of top features. A tabular list of the top features per decision class (ancestry) is shown in Table 1.

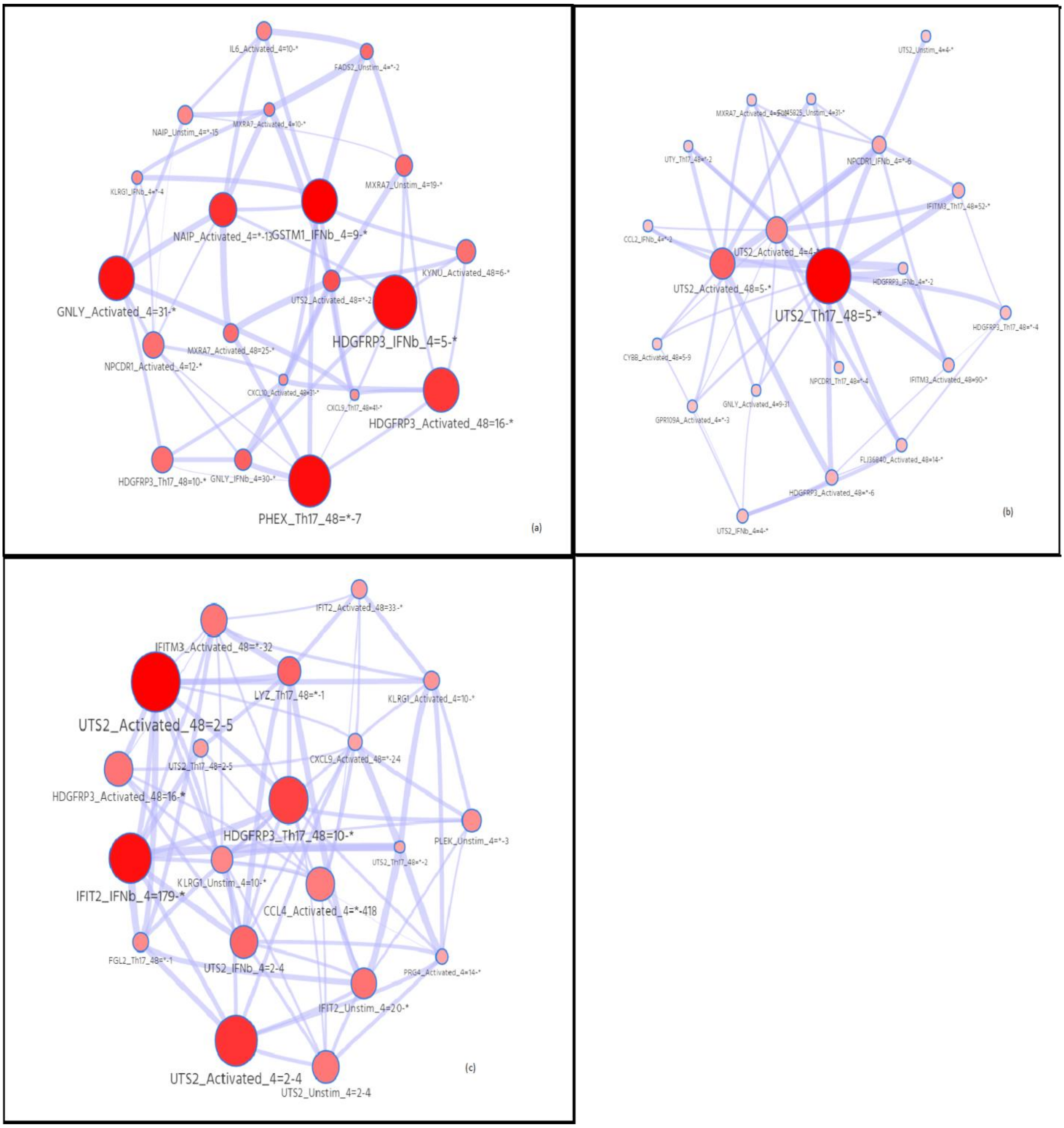


Figure 10: VisuNet representation of rule interactions. Networks of interaction for each of the three decision classes: (a) Afro, (b) Asian and (c) European. The nodes show genes with various treatments and their gene expression values. The edges are co-occurences of the nodes in rules (per decision).

Table 1: List of top 20 features in each of the three ancestry groups. The capitalized text before the first underscore is the gene name.

No.	African-American	Asian	European
1	GSTM1_IFNb_4=High	UTS2_Th17_48=High	UTS2_Activated_48=Mid
2	HDGFRP3_IFNb_4=High	UTS2_Activated_48=High	IFIT2_IFNb_4=High
3	PHEX_Th17_48=Low	UTS2_Activated_4=High	UTS2_Activated_4=Mid
4	GNLY_Activated_4=High	NPCDR1_IFNb_4=Low	HDGFRP3_Th17_48=High
5	NAIP_Activated_4=Low	HDGFRP3_Activated_48=Low	LYZ_Th17_48=Low
6	HDGFRP3_Activated_48=High	IFITM3_Th17_48=High	UTS2_IFNb_4=Mid
7	UTS2_Activated_48=Low	IFITM3_Activated_48=High	IFIT2_Unstim_4=High
8	GNLY_IFNb_4=High	UTS2_IFNb_4=High	HDGFRP3_Activated_48=High
9	MXRA7_Unstim_4=High	FLJ36840_Activated_48=High	UTS2_Unstim_4=Mid
10	FADS2_Unstim_4=Low	HDGFRP3_Th17_48=Low	IFITM3_Activated_48=Low
11	MXRA7_Activated_48=High	HDGFRP3_IFNb_4=Low	CCL4_Activated_4=Low
12	HDGFRP3_Th17_48=High	GPR109A_Activated_4=Low	KLRG1_Unstim_4=High
13	NPCDR1_Activated_4=High	MXRA7_Activated_4=Mid	FGL2_Th17_48=Low
14	KYNU_Activated_48=High	GNLY_Activated_4=Mid	PLEK_Unstim_4=Low
15	MXRA7_Activated_4=High	NPCDR1_Th17_48=Low	KLRG1_Activated_4=High
16	IL6_Activated_4=High	CCL2_IFNb_4=Low	UTS2_Th17_48=Mid
17	NAIP_Unstim_4=Low	UTS2_Unstim_4=High	IFIT2_Activated_48=High
18	KLRG1_IFNb_4=Low	CYBB_Activated_48=Mid	CXCL9_Activated_48=Low
19	CXCL10_Activated_48=High	FLJ45825_Unstim_4=High	UTS2_Th17_48=Low
20	CXCL9_Th17_48=High	UTY_Th17_48=Low	PRG4_Activated_4=High

In addition to visualizing interactions, VisuNet has the ability to annotate genomic data with KEGG and GO if the user provides an additional file mapping the node names to a gene name. We extracted the gene names ignoring the phenotypic variables, created the mapping file and fed it additionally with the rules from the top 100 features with the same settings as before. The result was an annotation with pathways which gave interesting overview of the data congruent to the first study. For instance, among the top pathways were Cytokine-cytokine receptor interaction and Chemokine signaling pathways shown in red text in Table 2, which had also been mentioned by the first study as key pathways involved. In addition, other than the common general GO terms, the classes had a lot of terms that related to immune response that would give the researcher a good starting place for delving further into the variations if need be.

Table 2: Top 10 metabolic pathways in each of the three ancestry groups annotated using KEGG data. Interesting pathways are highlighted in red text.

No.	African American	Asian	European
1	Cytokine-cytokine receptor interaction	AGE-RAGE signaling pathway in diabetic complications	Biosynthesis of unsaturated fatty acids
2	Toll-like receptor signaling pathway	Cytokine-cytokine receptor interaction	alpha-Linolenic acid metabolism
3	Chemokine signaling pathway	Malaria	Fatty acid metabolism
4	Legionellosis	Hematopoietic cell lineage	Toll-like receptor signaling pathway
5	NOD-like receptor signaling pathway	Chagas disease (American trypanosomiasis)	PPAR signaling pathway
6	Tryptophan metabolism	Herpes simplex infection	Chemokine signaling pathway
7	Influenza A	Influenza A	Cytokine-cytokine receptor interaction
8	Metabolic pathways	Rheumatoid arthritis	Salmonella infection
9	HIF-1 signaling pathway	NOD-like receptor signaling pathway	Metabolic pathways
10	AGE-RAGE signaling pathway in diabetic complications	TNF signaling pathway	NF-kappa B signaling pathway

In the first study, they state that choices between effector phenotypes are themselves modulated by the cytokine network, such as the reinforcement of the T_H17 identity through IL-23. These pathways also drive major immune-inflammatory diseases. Also, Pathogenic T_H1 or T_H17 cells have been implicated in rheumatoid arthritis, multiple sclerosis (MS), and inflammatory bowel disease (IBD), and T_H2-type responses in asthma and other atopic diseases. This can easily be seen in the GO terms and pathways listed in the VisuNet for the various classes without the need to go through external sources. Some interesting GO terms shown by VisuNet for the dataset are shown in Table 3.

Table 3: Select Gene Ontology terms by ancestry. These were some of the most common and interesting terms that represent the functions of the genes in the network.

No.	African American	Asian	European
1	inflammatory response	monocyte chemotaxis	inflammatory response
2	positive regulation of fibroblast proliferation	humoral immune response	cellular response to interferon-alpha
3	positive regulation of synaptic transmission, cholinergic	inflammatory response	proteolysis
4	immune response	innate immune response	response to virus
5	innate immune response	positive regulation of angiogenesis	cytokine-mediated signaling pathway
6	defense response to bacterium	response to drug	negative regulation of heart rate
7	cell proliferation	response to testosterone	regulation of blood pressure
8	chemokine activity	cytokine-mediated signaling pathway	negative regulation of insulin secretion
9	chemokine-mediated signaling pathway	immune response	response to drug
10	cytokine activity	response to interferon-gamma	immune response

The second study noted the overall importance of features in classification while not differentially examining their contributions to each class. Nevertheless, the top features and interactions between them hold a certain amount of truth although this would be more valuable *modulo* decision class. For instance, features representing the various states of the *UTS2* gene form the top features. While this may be true for the Asian and European populace, it does not reflect the distinctions per decision class. In VisuNet, however, it is clear that low-expression of the *UTS2* gene characterizes Afro group while High *UTS2* gene expression is the key characteristic of Asian group; this is also consistent with the mean gene expression levels shown by the first study. Also, other than UTS2, the top features vary considerably from one class to the other in VisuNet. It is oftentimes important for such studies which look at differential expressions that the researcher has an outlook of what features are important for what class and also possibly delve into what values of a feature are distinguishing e.g. high expression of a gene marks one class and low expression another etc.

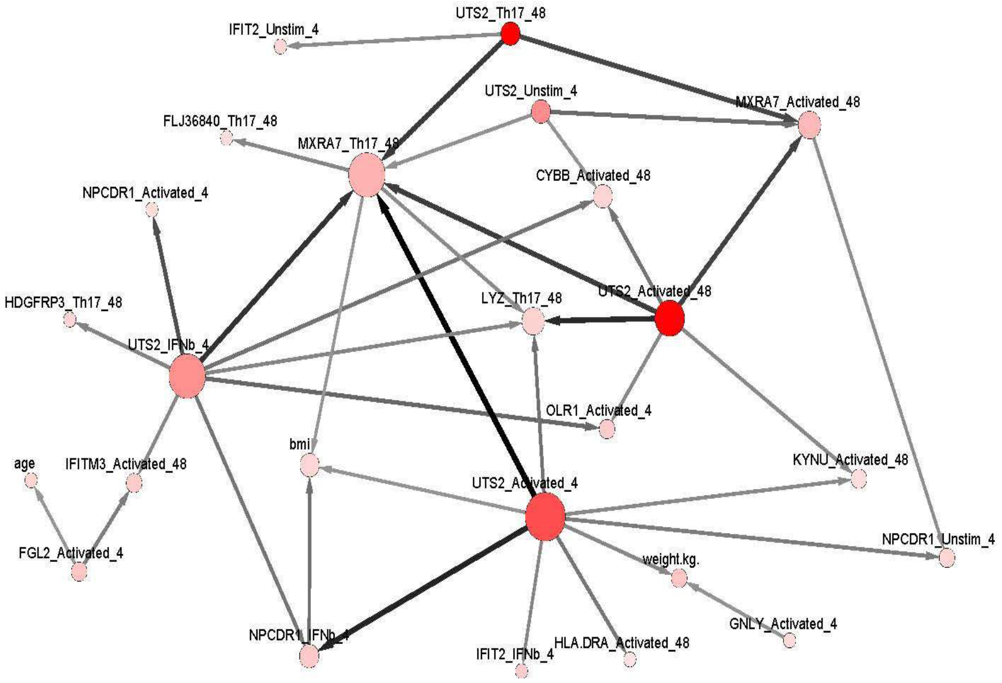


Figure 11: MCFS-ID graph showing top 50 features ordered by Relative Importance [54]**.**

5 Discussion

Any classification algorithm in which the trained model can be extracted as a set of rules with at least one measure of rule ranking can make use of VisuNet. This will include, but not limited to, Rough Sets (e.g. from ROSETTA) and similar fuzzy rule-based algorithms and Decision trees (e.g. from C4.5, CART). Such algorithms are easily understood by researchers, both in output and working, and are quite useful in most classification cases where the classifier has to be opened up and features making up the rules in the trained model investigated [54]. Just as it is with other statistical inferencing techniques such as logistic regression, it is important that the researcher considers not only the ranking of the features based on some importance measure but also to identify features that co-operate in the classification of an object. This is definitely a major strength of VisuNet: visualizing these interactions.

Additionally, for genomic data, VisuNet provides the ability to plug in a mapping file to link features to genes and populates annotations from two biological networks – KEGG pathways and GO terms – in each class in an interactive way. The value of information for biological networks cannot be stressed enough. Scientists no longer consider individual molecules since the cell is one complex network of co-acting and interacting molecules. It is therefore important to study the system as a whole and the interactions within it. There are two main methods for identification of network structures: bottom-up (knowledge-driven) and top-down (data-driven) approaches [72].

5.1 Biological Networks

KEGG [73] Pathways and Gene Ontology [74] terms use a bottom-up approach for construction of the excellent and growing database of various biological networks. KEGG constructs excellent interaction diagrams of metabolic pathways for known cases, mostly curated using literature but also combining homology-searches for functional inference. GO is a resource that supplies information about gene product function using ontologies to represent biological knowledge. Both of these resources give references citing relevant sources of information making it easier for the researcher to look further than the network diagram. It is on the premise of such importance that VisuNet is built: the incorporation of biological networks in the tool allows the researcher a wealth of relevant information at the click of a button. It is important to reiterate clearly that network presented by VisuNet does not infer causality (hence no direction) even though any interactions that may occur in the same order (in VisuNet versus in KEGG Pathways, for instance) definitely would be quite interesting to investigate.

5.2 Challenge of feature selection

Feature selection is used in classification to reduce the number of features that a classifier uses based on some heuristic. Independent of the method used, feature selection aims to pick the most informative set of features from the universe of features. One example paradigm in feature selection is to pick features that are highly correlated to the decision class and less with each other [75]. This can lead to overshadowing of equally important but correlated features and hence for the interest of this study, loss of vital information. If two features are almost equally important but correlated, one may be lost and the other retained by the feature selection algorithm. After a classifier is built, we have fewer correlating items than in the original dataset. Some algorithms, such as Random Reducts proposed by [76] in which features are picked randomly and Reducts calculated from them present an option that would prevent complete overshadowing. We therefore propose that the filtering process be done with overshadowing in consideration. For instance, the classification can be done without feature selection if it is computationally plausible.

6 Conclusion

In conclusion, VisuNet offers not only an aesthetically-pleasant, highly interactive and natural layout of the interactions between features but in addition, a user looking into a specific area of computational biology will be able to at first glance answer several questions: What are the highest ranking features in terms of contribution to top rules? What are the strongest interactions between features per decision class? What gene is represented by this feature? What pathways and terms does the gene participate in? What features in my current view are in this pathway? And many more such questions. The foraging through databases is reduced significantly and quick initial revelations could be made efficiently.

7 Future work

There is definitely a lot that could be achieved with VisuNet as-is but there is always need for continuous improvement. One of the key things would be to improve performance for large rule files and even greater fluidity of the UI. Also, some scripts to update the KEGG and GO term mirrors periodically have been provided but will need improvement to prevent any disruption of service. The ability to store user settings by allowing them to log in, and keep data for some period is also worth discussing although privacy and security is of essence in such cases.

8 Acknowledgements

I would not have done this without the kind assistance of Husen M. Umer who reviewed the draft and gave great insight into this report. I would also like to extend my sincere gratitude to Professor Jan Komorowski and the team at Komorowski's Lab which has supported me countless times and endured multiple interruptions patiently. Finally, I am quite thankful to

my colleague and office roommate, Nicholas Baltzer, who provided enormous support on algorithm design and was a very reliable brainstorming partner.

9 References

1. Kaelbling LP, Littman ML, Moore AW. Reinforcement Learning: A Survey. J. Artif. Int. Res. 1996;4:237–85.

2. Zhu X. Semi-Supervised Learning. In: Sammut C, Webb GI, editors. Encyclopedia of Machine Learning [Internet]. Springer US; 2011 [cited 2016 Mar 5]. p. 892–7. Available from: http://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_749

3. Hady MFA, Schwenker F. Semi-supervised Learning. In: Bianchini M, Maggini M, Jain LC, editors. Handbook on Neural Information Processing [Internet]. Springer Berlin Heidelberg; 2013 [cited 2016 Mar 5]. p. 215–39. Available from: http://link.springer.com/chapter/10.1007/978-3-642-36657-4_7

4. Mitra S, Datta S, Perkins T, Michailidis G. Introduction to Machine Learning and Bioinformatics. CRC Press; 2008.

5. Kislyuk A, Bhatnagar S, Dushoff J, Weitz JS. Unsupervised statistical clustering of environmental shotgun sequences. BMC Bioinformatics. 2009;10:316.

6. Wong G, Chan J, Kingwell BA, Leckie C, Meikle PJ. LICRE: unsupervised feature correlation reduction for lipidomics. Bioinformatics. 2014;30:2832–3.

7. Kotsiantis SB. Supervised Machine Learning: A Review of Classification Techniques. Informatica [Internet]. 2007 [cited 2016 Mar 5];31. Available from: http://www.informatica.si/index.php/informatica/article/view/148

8. Cortes C, Vapnik V. Support-Vector Networks. Machine Learning. 1995. p. 273–97.

9. Modak S, Sharma S, Prabhakar P, Yadav A, Jayaraman VK. Application of Support Vector Machines in Fungal Genome and Proteome Annotation. In: Gupta VK, Tuohy MG, Ayyachamy M, Turner KM, O'Donovan A, editors. Laboratory Protocols in Fungal Biology [Internet]. Springer New York; 2013 [cited 2016 Mar 5]. p. 565–77. Available from: http://link.springer.com/chapter/10.1007/978-1-4614-2356-0_56

10. Du W, Cheung H, Johnson CA, Goldberg I, Thambisetty M, Becker K. A longitudinal support vector regression for prediction of ALS score. 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2015. p. 1586–90.

11. Balfer J, Bajorath J. Visualization and Interpretation of Support Vector Machine Activity Predictions. J. Chem. Inf. Model. 2015;55:1136–47.

12. Fung G, Sandilya S, Rao RB. Rule Extraction from Linear Support Vector Machines. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining [Internet]. New York, NY, USA: ACM; 2005 [cited 2016 Mar 1]. p. 32–40. Available from: http://doi.acm.org/10.1145/1081870.1081878

13. Castro JL, Flores-Hidalgo LD, Mantas CJ, Puche JM. Extraction of fuzzy rules from support vector machines. Fuzzy Sets and Systems. 2007;158:2057–77.

14. Martens D, Baesens B, Van Gestel T, Vanthienen J. Comprehensible credit scoring models using rule extraction from support vector machines. European Journal of Operational Research. 2007;183:1466–76.

15. Barakat N, Bradley AP. Rule extraction from support vector machines: A review. Neurocomputing. 2010;74:178–90.

16. Wang S-C. Artificial Neural Network. Interdisciplinary Computing in Java Programming [Internet]. Springer US; 2003 [cited 2016 Mar 5]. p. 81–100. Available from: http://link.springer.com/chapter/10.1007/978-1-4615-0377-4_5

17. Huesken D, Lange J, Mickanin C, Weiler J, Asselbergs F, Warner J, et al. Design of a genome-wide siRNA library using an artificial neural network. Nat Biotech. 2005;23:995–1001.

18. Baldi P, Brunak S. Bioinformatics : The Machine Learning Approach [Internet]. Cambridge, MA, USA: MIT Press; 2001 [cited 2016 Mar 7]. Available from: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10225255

19. Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, et al. Machine learning in bioinformatics. Brief Bioinform. 2006;7:86–112.

20. Benitez JM, Castro JL, Requena I. Are artificial neural networks black boxes? IEEE Transactions on Neural Networks. 1997;8:1156–64.

21. Kulluk S, Özbakır L, Baykasoğlu A. Fuzzy DIFACONN-miner: A novel approach for fuzzy rule extraction from neural networks. Expert Systems with Applications. 2013;40:938–46.

22. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–44.

23. Chandra B, Sharma RK. Fast learning in Deep Neural Networks. Neurocomputing. 2016;171:1205–15.

24. Leung MKK, Xiong HY, Lee LJ, Frey BJ. Deep learning of the tissue-regulated splicing code. Bioinformatics. 2014;30:i121–9.

25. Xiong HY, Alipanahi B, Lee LJ, Bretschneider H, Merico D, Yuen RKC, et al. The human splicing code reveals new insights into the genetic determinants of disease. Science. 2015;347:1254806.

26. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. J. Chem. Inf. Comput. Sci. 2003;43:1947–58.

27. Breiman L. Random Forests. Machine Learning. 2001;45:5–32.

28. Díaz-Uriarte R, Alvarez de Andrés S. Gene selection and classification of microarray data using random forest. BMC Bioinformatics. 2006;7:3.

29. Goldstein BA, Hubbard AE, Cutler A, Barcellos LF. An application of Random Forests to a genome-wide association dataset: Methodological considerations & new findings. BMC Genetics. 2010;11:49.

30. Nguyen T-T, Huang JZ, Wu Q, Nguyen TT, Li MJ. Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests. BMC Genomics. 2015;16:S5.

31. Komorowski EØJ. ROSETTA -- A Rough Set Toolkit for Analysis of Data. ResearchGate [Internet]. 1997 [cited 2016 Mar 3]; Available from: https://www.researchgate.net/publication/2259892_ROSETTA_--_A_Rough_Set_Toolkit_for_Analysis_of_Data

32. Øhrn A, Komorowski J, Skowron A, Synak P. The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets - The ROSETTA System. 1998.

33. Kingsford C, Salzberg SL. What are decision trees? Nat Biotech. 2008;26:1011–3.

34. Murthy SK. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery. 1998;2:345–89.

35. Darnell SJ, Page D, Mitchell JC. An automated decision-tree approach to predicting protein interaction hot spots. Proteins. 2007;68:813–23.

36. Mestizo Gutiérrez SL, Herrera Rivero M, Cruz Ramírez N, Hernández E, Aranda-Abreu GE. Decision trees for the analysis of genes involved in Alzheimer׳s disease pathology. Journal of Theoretical Biology. 2014;357:21–5.

37. Motsinger-Reif AA, Deodhar S, Winham SJ, Hardison NE. Grammatical evolution decision trees for detecting gene-gene interactions. BioData Mining. 2010;3:8.

38. Pontac M, Bourrier T, Le Heron C, Rocher F, Marquette C-H, Leroy S. Hypersensibilités aux AINS : phénotypes cliniques et arbre décisionnel. Revue Française d’Allergologie. 2015;55:392–400.

39. Naegeli H, Sugasawa K. The xeroderma pigmentosum pathway: Decision tree analysis of DNA quality. DNA Repair. 2011;10:673–83.

40. Surucu M, Shah KK, Mescioglu I, Roeske JC, Small W, Choi M, et al. Decision Trees Predicting Tumor Shrinkage for Head and Neck Cancer Implications for Adaptive Radiotherapy. Technol Cancer Res Treat. 2016;15:139–45.

41. Dramiński M, Kierczak M, Koronacki J, Komorowski J. Monte Carlo feature selection and interdependency discovery in supervised classification. Springer; 2010 [cited 2016 Mar 7]. Available from: http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A274118&dswid=6966

42. Quinlan JR. C4.5: Programs for Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1993.

43. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. CRC press; 1984.

44. Pawlak Z. Rough sets. International Journal of Computer and Information Sciences. 1982;11:341–56.

45. Birney E, Stamatoyannopoulos JA, Dutta A, Guigó R, Gingeras TR, Margulies EH, et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature. 2007;447:799–816.

46. Khaliq Z, Leijon M, Belák S, Komorowski J. A complete map of potential pathogenicity markers of avian influenza virus subtype H5 predicted from 11 expressed proteins. BMC Microbiol. 2015;15:128.

47. Rzepiński T. Randomized controlled trials versus rough set analysis: two competing approaches for evaluating clinical data. Theor Med Bioeth. 2014;35:271–88.

48. Sahiner A, Yigit T, Sahiner A, Yigit T. A Study of Rough Set Approach in Gastroenterology. Computational and Mathematical Methods in Medicine, Computational and Mathematical Methods in Medicine. 2013;2013, 2013:e782049.

49. Duda RO, Hart PE, Stork DG. Pattern Classification (2Nd Edition). Wiley-Interscience; 2000.

50. Zhao Z, Liu H. Searching for Interacting Features in Subset Selection. Intell. Data Anal. 2009;13:207–28.

51. Yu L, Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution. 2003. p. 856–63.

52. Amiri F, Rezaei Yousefi M, Lucas C, Shakery A, Yazdani N. Mutual information-based feature selection for intrusion detection systems. Journal of Network and Computer Applications. 2011;34:1184–99.

53. Wu R, Pruitt Z, Runkle M, Meyer K, Scerif G, Aslin R. Feature correlation guidance in category visual search. Journal of Vision. 2015;15:926.

54. Dramiński M, Dąbrowski MJ, Diamanti K, Koronacki J, Komorowski J. Discovering Networks of Interdependent Features in High-Dimensional Problems. In: Japkowicz N, Stefanowski J, editors. Big Data Analysis: New Algorithms for a New Society [Internet]. Springer International Publishing; 2015 [cited 2016 Mar 3]. p. 285–304. Available from: http://link.springer.com/chapter/10.1007/978-3-319-26989-4_12

55. Liu H, Li M, Zhao J, Mo Y. An Effective Feature Selection Method Using Dynamic Information Criterion. In: Deng H, Miao D, Lei J, Wang FL, editors. Artificial Intelligence and Computational Intelligence [Internet]. Springer Berlin Heidelberg; 2011 [cited 2016 Mar 5]. p. 450–5. Available from: http://link.springer.com/chapter/10.1007/978-3-642-23881-9_59

56. Ye CJ, Feng T, Kwon H-K, Raj T, Wilson MT, Asinovski N, et al. Intersection of population variation and autoimmunity genetics in human T cell activation. Science. 2014;345:1254665.

57. Dramiński M, Rada-Iglesias A, Enroth S, Wadelius C, Koronacki J, Komorowski J. Monte Carlo feature selection for supervised classification. Bioinformatics. 2008;24:110–7.

58. Hoffman DL, Novak TP, Venkatesh A. Has the Internet become indispensable? Communications of the ACM. 2004;47:37–42.

59. Lesne A. Complex Networks: from Graph Theory to Biology. Lett Math Phys. 2006;78:235–62.

60. Balakrishnan R, Ranganathan K. A Textbook of Graph Theory [Internet]. New York, NY: Springer New York; 2012 [cited 2016 Mar 5]. Available from: http://link.springer.com/10.1007/978-1-4614-4529-6

61. Bornelöv S, Marillet S, Komorowski J. Ciruvis: a web-based tool for rule networks and interaction detection using rule-based classifiers. BMC Bioinformatics. 2014;15:139.

62. Hofmann H, Siebes APJM, Wilhelm AFX. Visualizing Association Rules with Interactive Mosaic Plots. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and

Data Mining [Internet]. New York, NY, USA: ACM; 2000 [cited 2016 Mar 3]. p. 227–35. Available from: http://doi.acm.org/10.1145/347090.347133

63. Hahsler M, Chelluboina S. Visualizing association rules: Introduction to the R-extension package arulesViz. R project module. 2011;223–38.

64. Hahsler M, Chelluboina S. Visualizing Association Rules in Hierarchical Groups. In 42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface. 2011.

65. Hahsler M, Buchta C, Gruen B, Hornik K. arules: Mining Association Rules and Frequent Itemsets [Internet]. 2015. Available from: http://CRAN.R-project.org/package=arules

66. Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Softw: Pract. Exper. 1991;21:1129–64.

67. Dong W, Fu X, Xu G, Huang Y. An improved force-directed graph layout algorithm based on aesthetic criteria. Comput. Visual Sci. 2014;16:139–49.

68. Gansner ER, North SC. Improved Force-Directed Layouts. In: Whitesides SH, editor. Graph Drawing [Internet]. Springer Berlin Heidelberg; 1998 [cited 2016 Mar 5]. p. 364–73. Available from: http://link.springer.com/chapter/10.1007/3-540-37623-2_28

69. Franz M, Lopes CT, Huck G, Dong Y, Sumer O, Bader GD. Cytoscape.js: a graph theory library for visualisation and analysis. Bioinformatics. 2016;32:309–11.

70. Dubey P, Shingare A, Inamdar V. A Force Directed Layout Algorithm for Biological Networks. International Journal of Computer Applications. 2015;120:42–7.

71. Mudunuri U, Che A, Yi M, Stephens RM. bioDBnet: the biological database network. Bioinformatics. 2009;25:555–6.

72. Chen L, Wang R-S, Zhang X-S. Biomolecular Networks : Methods and Applications in Systems Biology [Internet]. Hoboken, NJ, USA: John Wiley & Sons; 2009 [cited 2016 Feb 24]. Available from: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10315655

73. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M. KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res. 2012;40:D109–14.

74. Gene Ontology Consortium. Gene Ontology Consortium: going forward. Nucleic Acids Res. 2015;43:D1049–56.

75. Hall MA. Correlation-based Feature Selection for Machine Learning. 1999.

76. Marcin Kruczyk NB. Random Reducts: A Monte Carlo Rough Set-based Method for Feature Selection in Large Datasets. Fundamenta Informaticae. 2013;127:273–88.

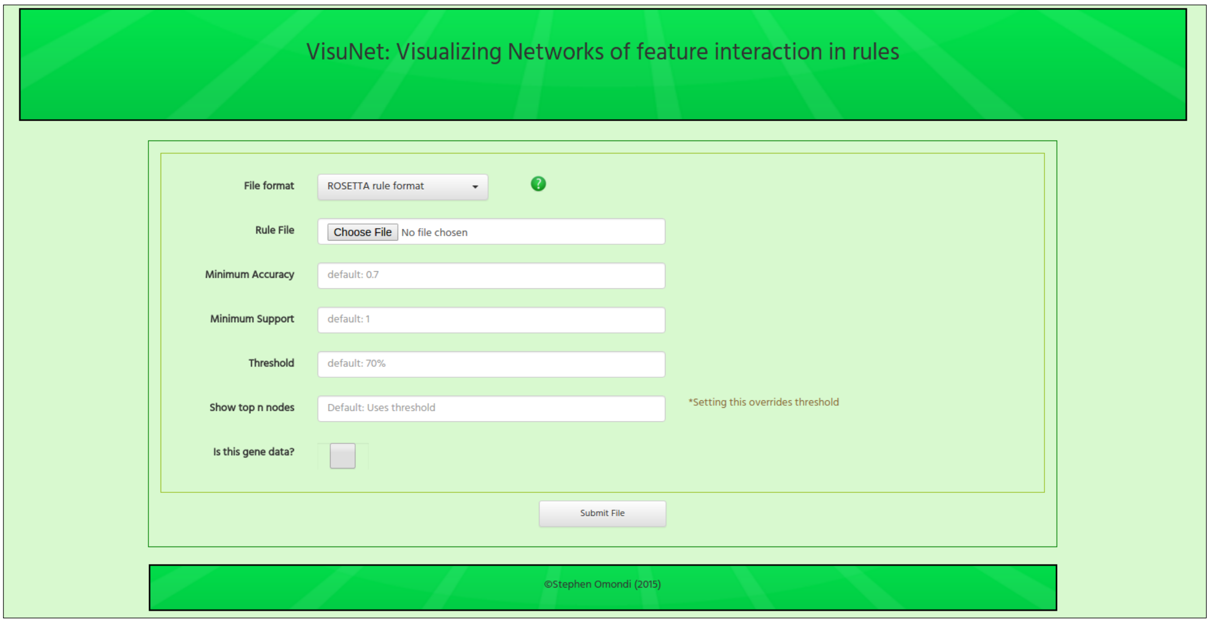
APPENDIX I: User's Manual

Introduction

VisuNet is an interactive web-based application that visualizes interactions between features in rule-based classifiers in a network layout. Optionally, it allows the user to add a mapping file for genomic data and annotates the network with biological information from Kyoto Encyclopedia of Genes and Genomes (KEGG) Pathways as well as Gene Ontology (GO) terms. You can find more information about KEGG and GO at http://www.kegg.jp/kegg/ and http://geneontology.org/ respectively.

PART I: The Home Page

Below is a screenshot of the application's home page. The next sections will detail the file formats used as input and, the filtering options provided on the screen. Click the Choose file or Browse (IE/Firefox) button next to the Rule File label and select the file containing rules that you want to visualize.



VisuNet: Visualizing Networks of feature interaction in rules

File format: ROSETTA rule format

Rule File: Choose File No file chosen

Minimum Accuracy: default: 0.7

Minimum Support: default: 1

Threshold: default: 70%

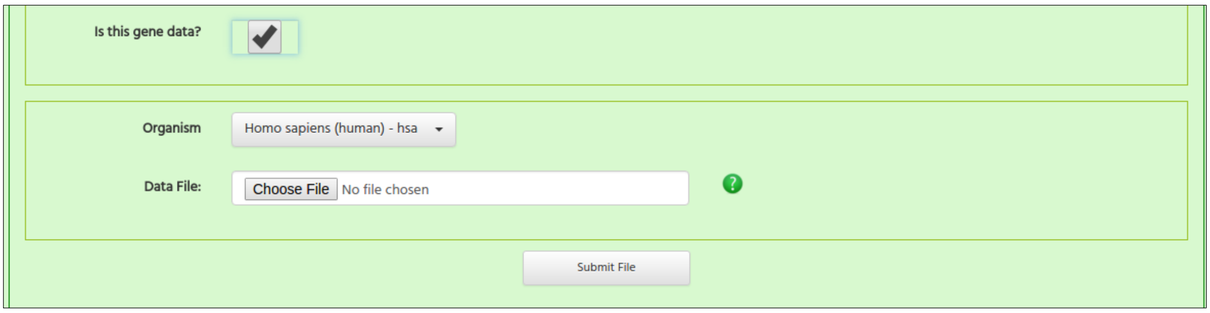
Show top n nodes: Default: Uses threshold *Setting this overrides threshold

Is this gene data? □

Submit File

©Stephen Omondi (2015)

If you have genic data, check the **"Is this gene data"** checkbox. This toggles a section with the **organism** and **Mapping file** labels. Select the organism in study and select a file for mapping. The file formats are described in the next session. Click the submit button to load the network of interactions.



Is this gene data? ☑

Organism: Homo sapiens (human) - hsa

Data File: Choose File No file chosen

Submit File

You will see a Rubik's cube image while the page is loading. Wait for the loading to complete and you will be automatically redirected to the next page. In case of errors, the application will show an appropriate error message.

Section 1: Input File Types

The application accepts two types of input files for the rules: ROSETTA rule file or a Line-By-Line text file. The ROSETTA file format (.ros) can be found if you use the ROSETTA rule-based classification tool found at http://www.lcb.uu.se/tools/rosetta/resources.php. The linked website also provides a manual on how to use ROSETTA to get a classifier. An example file for the ROSETTA format is as below:

```
ER(1) AND (1) AND FABP(+) => Site(Ov)
Supp. (LHS) = [1 object(s)]
Supp. (RHS) = [1 object(s)]
Acc.  (RHS) = [1]
Cov.  (LHS) = [0.0042735]
Cov.  (RHS) = [0.0384615]
Stab. (LHS) = [1]
Stab. (RHS) = [1]
```

The Line-by-Line format presents each rule in the model as a single line. The features are separated by commas and **should NOT have commas within the text** since the comma is used as a separator e.g. age=[32,*], MXRA7_Activated_48=[26,*] are bad feature names/values.

```
age=32-*,MXRA7_Activated_48=26-*,      Afro  0.745 47
weight.kg.=78.2-*,MXRA7_Activated_48=26-*,  Afro  0.644 45
CYBB_Activated_48=9-*,MXRA7_Activated_48=26-*,   Afro  0.612 49
UTS2_Activated_4=*-2,CYBB_Activated_48=9-*, Afro  0.579 57
MXRA7_Activated_48=26-*,UTS2_Th17_48=*-2,   Afro  0.545 66
```

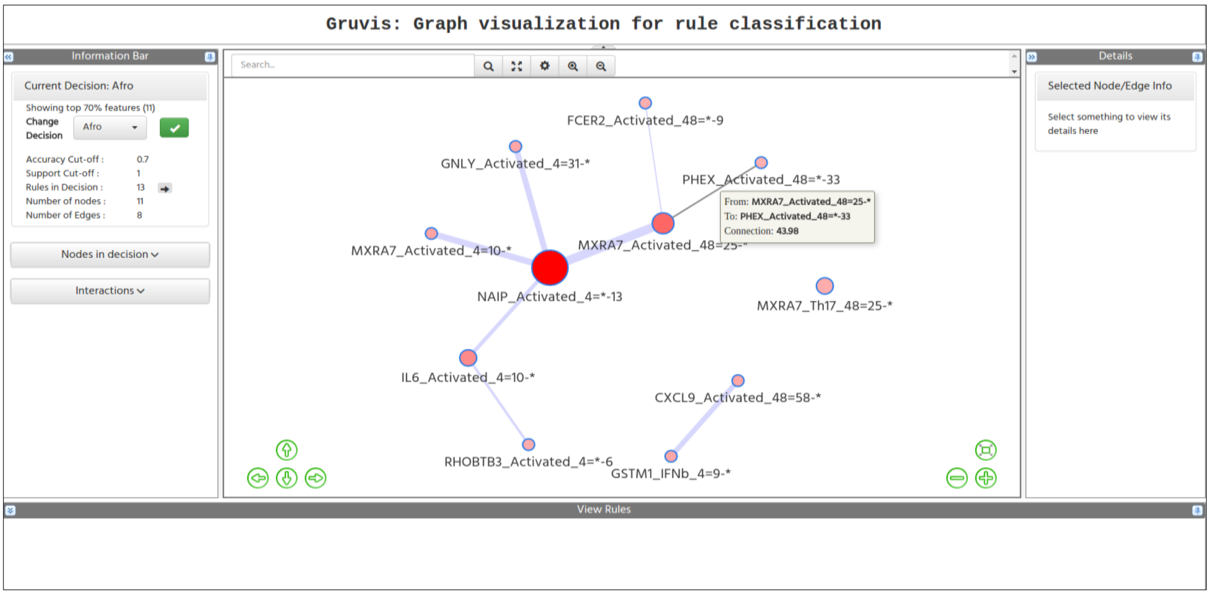
Section 2: Pre-Filtering

The number of nodes generated can be so many and hence it is important to pre-filter the data to avoid clutter. The most straight forward and highly advised method to begin is to set the number of nodes to be displayed in the "**Show n nodes"** field. You can also set the threshold to only show top x percent of the nodes e.g. setting 70%, which is the default, shows the top 70% of the nodes.

You can additionally filter by accuracy or support which excludes all rules with accuracy or support below the specified values in the downstream analysis.

PART II: Visualizing the network

The next page after submission shows the network diagram at the centre. Three collapsible panels on the left (Information Bar), right (Details) and bottom (View Rules) give additional information. If on loading the network keeps going in a circular manner, kindly wait for it to stabilize and get an optimal layout. This could take a few minutes depending on the number of nodes in the diagram.



Gruvis: Graph visualization for rule classification

Information Bar

Current Decision: Afro

Showing top 70% features (11)

Change Decision: Afro

Accuracy Cut-off : 0.7
Support Cut-off : 1
Rules in Decision : 13
Number of nodes : 11
Number of Edges : 8

From: MXRA7_Activated_48=25-*
To: PHEX_Activated_48=*-33
Connection: 43.98

Details

Selected Node/Edge Info

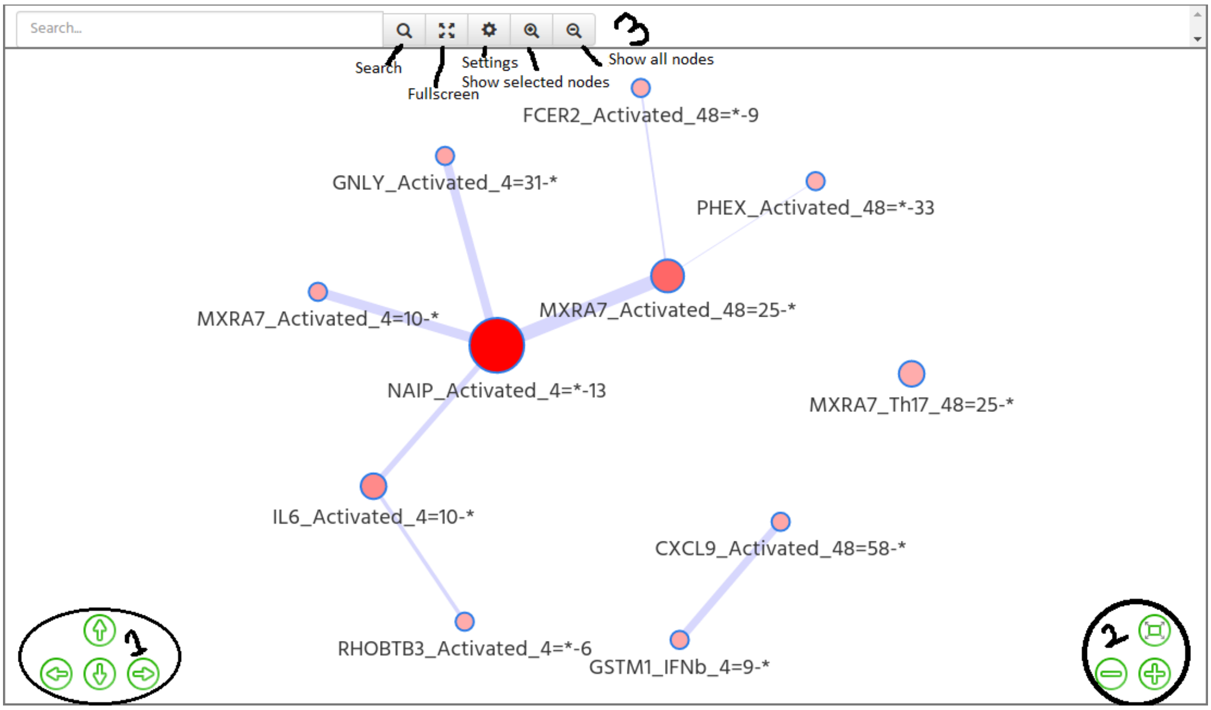
Select something to view its details here

Section 1: The network diagram

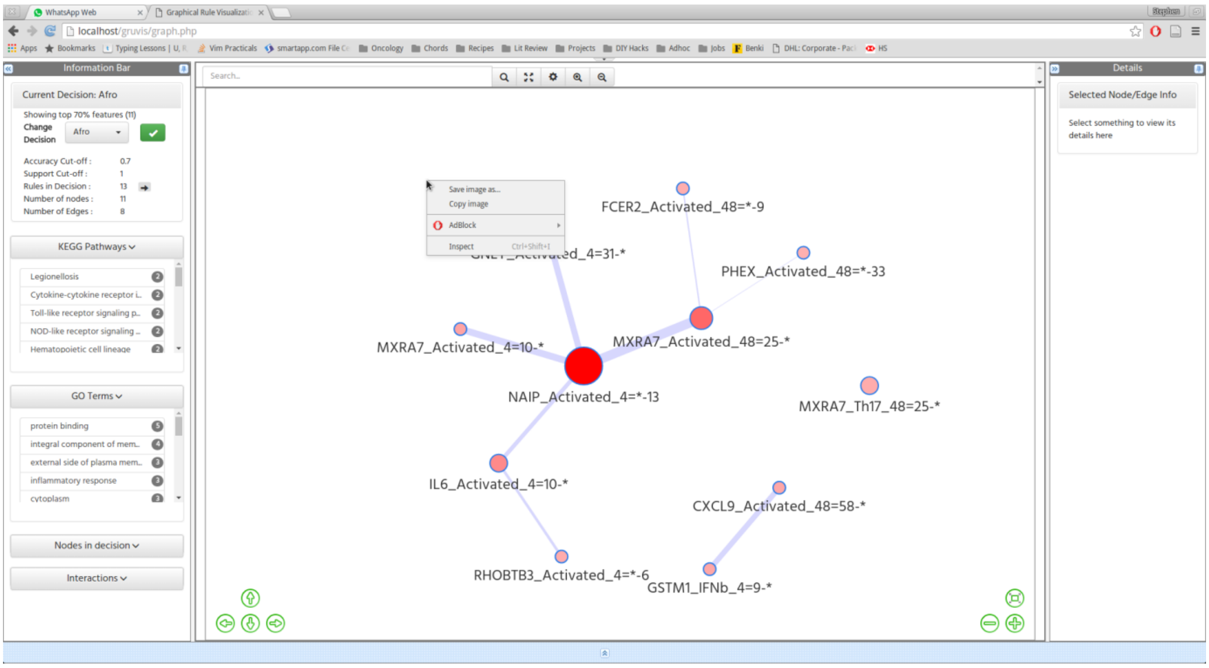
The network is drawn on the centre of the page as shown below. The green group of buttons on bottom left (1) enable you to move the network diagram in all directions to adjust its position. The window can also be dragged using the mouse button (click, hold and move). The ones on the bottom right corner (2) enable you to fit the network diagram into the window and also zoom in/out. The zooming can be done using the scroll button on the mouse too.

There is also a toolbar (3) above the network diagram with search box (search the nodes by part of feature name) and other buttons labelled in the screenshot below. To search, you type the search term into the toolbar and press the "Enter" key or click Search button. All matching nodes are highlighted and their information shown on the details pane and the View Rules window. It is possible to collapse all the panels so as to enlarge the view of the centre pane where the network diagram is. Additionally the full-screen button allows the browser to go into full-screen mode giving an even larger view port. The settings window allows the user to change some settings like colors and borders. The Show selected nodes button on the toolbar removes all non-selected nodes and only shows the selected nodes and interactions between them (if any). The Show all nodes button restores all the nodes into view.

The circles in the network diagram are nodes/conditions. A condition is a feature/value pair, for example "GNLY_Activated_4=31-*" in the screenshot below. Nodes are colored red with varying intensity. The best quality node is the brightest and strongest in color intensity e.g. NAIP_Activated_4 in the screenshot below. The lines connecting the nodes (edges) vary in thickness depending on the number of occurrences of the pair of features in rules and the quality of those rules. The strongest interaction in the diagram (For example, between NAIP_Activated_4=*-13 and MXRA7_Activated_48=25-*) has the thickest width.



Users can export the image in PNG format by right clicking the network window and selecting save image as or view image (opens in a new window/tab) buttons.


Current Decision: Afro

Showing top 70% features (11)

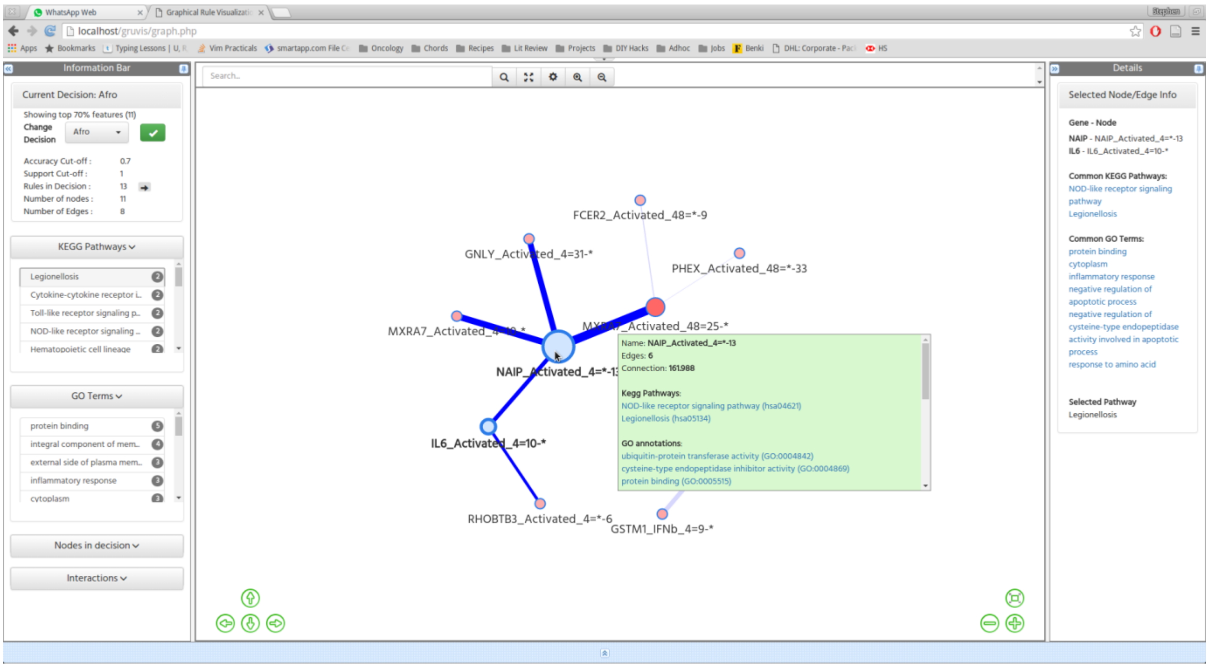
Change Decision: Afro

Accuracy Cut-off : 0.7
Support Cut-off : 1
Rules in Decision : 13
Number of nodes : 11
Number of Edges : 8

Selected Node/Edge Info

Select something to view its details here

Hovering over the nodes or edges also shows the details of the node/edge as shown below.



Information Bar

Current Decision: Afro

Showing top 70% features (11)

Change Decision: Afro

Accuracy Cut-off : 0.7
Support Cut-off : 1
Rules in Decision : 13
Number of nodes : 11
Number of Edges : 8

KEGG Pathways

Legionellosis	2
Cytokine-cytokine receptor i...	2
Toll-like receptor signaling p...	2
NOD-like receptor signaling ...	2
Hematopoietic cell lineage	2

GO Terms

protein binding	5
integral component of mem...	4
external side of plasma mem...	3
inflammatory response	3
cytoplasm	3

Nodes in decision

Interactions

Name: NAIP_Activated_4=*-13
Edges: 6
Connection: 161.988

Kegg Pathways:
NOD-like receptor signaling pathway (hsa04621)
Legionellosis (hsa05134)

GO annotations:
ubiquitin-protein transferase activity (GO:0004842)
cysteine-type endopeptidase inhibitor activity (GO:0004869)
protein binding (GO:0005515)

Details

Selected Node/Edge Info

Gene - Node
NAIP - NAIP_Activated_4=*-13
IL6 - IL6_Activated_4=10-*

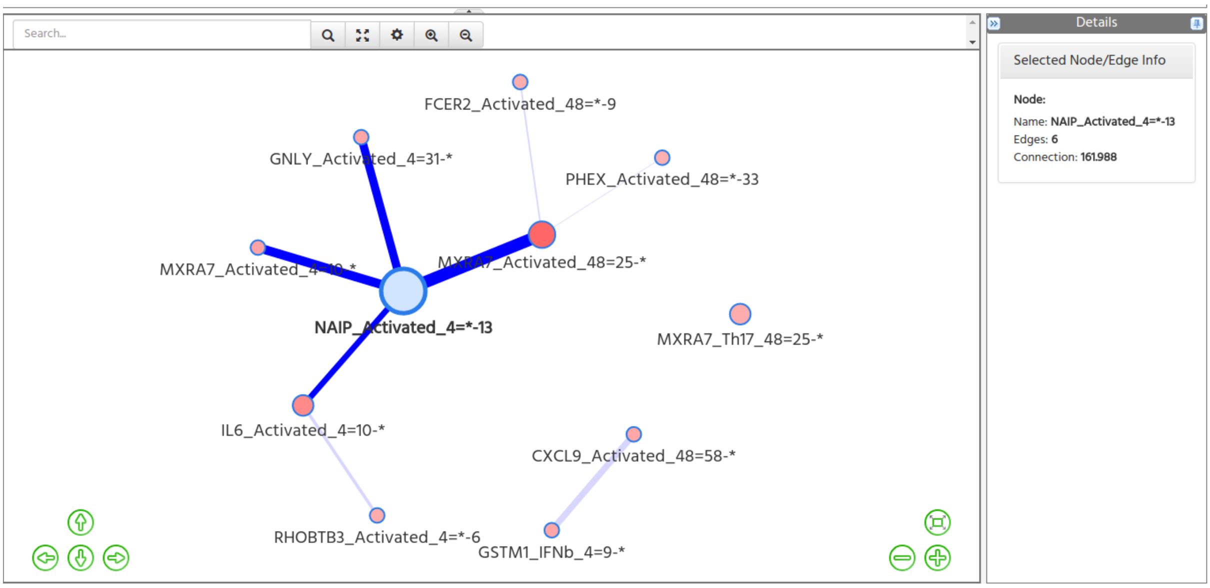
Common KEGG Pathways:
NOD-like receptor signaling pathway
Legionellosis

Common GO Terms:
protein binding
cytoplasm
inflammatory response
negative regulation of apoptotic process
negative regulation of cysteine-type endopeptidase activity involved in apoptotic process
response to amino acid

Selected Pathway
Legionellosis

Section 2: The side panels

When nodes or edges are selected, the details panel on the right shows information about the node. The node is highlighted in light blue with the edges emanating from it in a strong blue color as shown below. In the basic scenario (without a mapping file) there is not much information displayed. For nodes, the name, number of edges and a quality measure (connection) are shown. For edges, the 'from' and 'to' node names as well as connection are shown. The 'from' and 'to' texts do not indicate direction but mere connectedness.

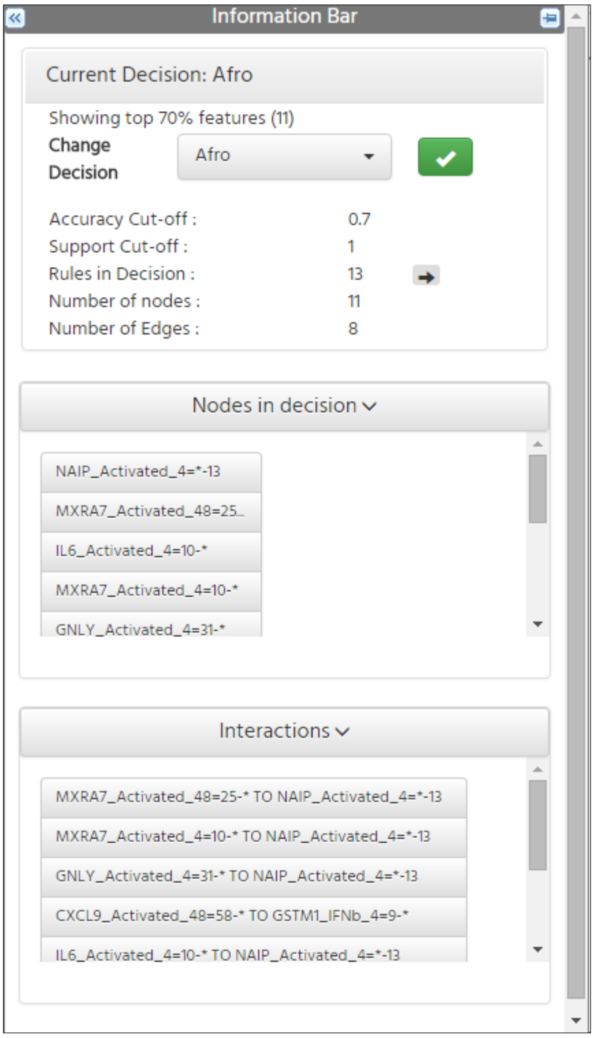


Details

Selected Node/Edge Info

Node:
Name: NAIP_Activated_4=*-13
Edges: 6
Connection: 161.988

The information panel on the left hand side has three basic sections as shown below:



Current Decision: Afro

Showing top 70% features (11)

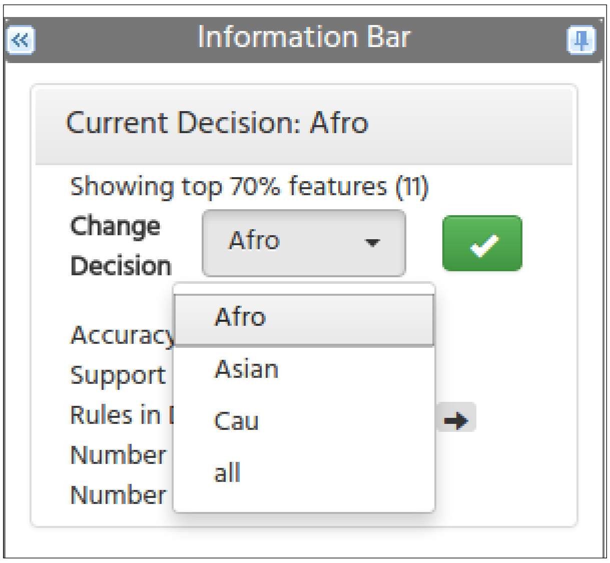
Change Decision: Afro ✓

Accuracy Cut-off :	0.7
Support Cut-off :	1
Rules in Decision :	13
Number of nodes :	11
Number of Edges :	8

NAIP_Activated_4=*-13
MXRA7_Activated_48=25_
IL6_Activated_4=10-*
MXRA7_Activated_4=10-*
GNLY_Activated_4=31-*

MXRA7_Activated_48=25-* TO NAIP_Activated_4=*-13
MXRA7_Activated_4=10-* TO NAIP_Activated_4=*-13
GNLY_Activated_4=31-* TO NAIP_Activated_4=*-13
CXCL9_Activated_48=58-* TO GSTM1_IFNb_4=9-*
IL6_Activated_4=10-* TO NAIP_Activated_4=*-13

The first section displays the information about the outcome (decision class) being displayed currently and the settings done in the home page. It also shows basic statistics about the displayed network. In this section, the user can change the decision outcome by selecting the item on the drop down menu and clicking the GO button adjacent to it.



Current Decision: Afro

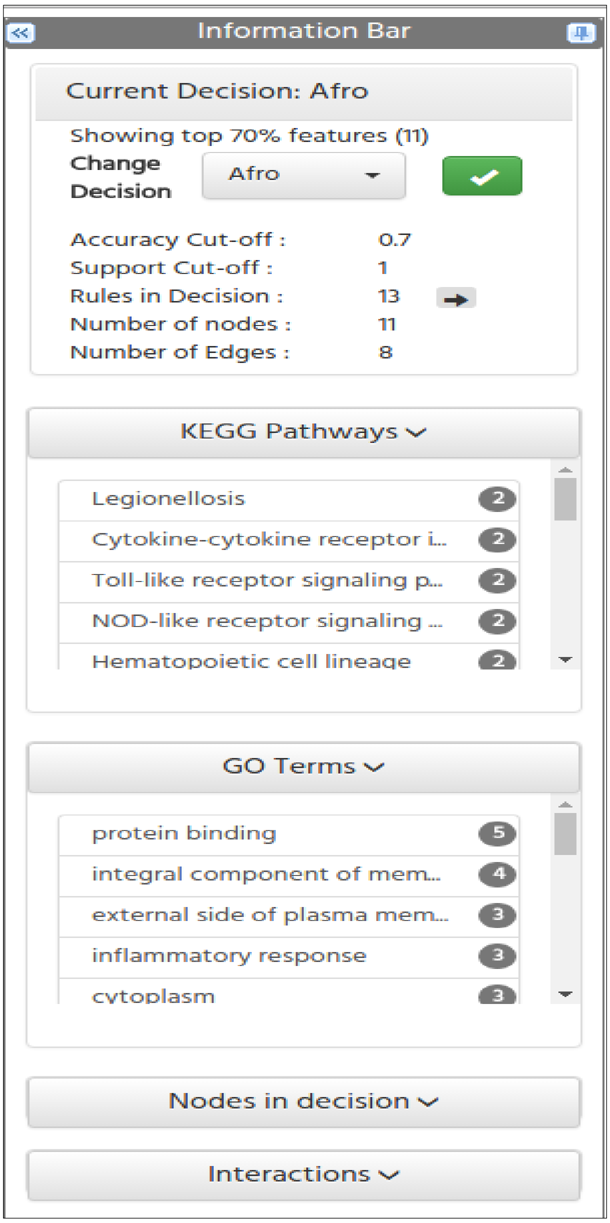
Showing top 70% features (11)

Change Decision: Afro ✓

The next two sections show the list of nodes in the network and the interactions respectively. Both are ordered by connection in descending order (e.g. top node first). Clicking the nodes selects it in the network diagram and displays its details in the detail pane. The rules in which the node is found are also shown in the view rules pane. The same goes for clicking the interactions.

Section 3: Visualizing biological information

If the user inputs a mapping file, VisuNet annotates the nodes with data of biological networks specifically from KEGG Metabolic pathways and GO terms. Two additional sections are added to the information pane as shown below . The numbers next to the name of the pathway or GO term is the number of nodes in the visible network diagram that are found in the pathway or are annotated with GO term respectively. Clicking the list items selects the nodes and displays information about them similar to selecting a node.



Information Bar

Current Decision: Afro

Showing top 70% features (11)

Change Decision: Afro ✓

Accuracy Cut-off : 0.7
Support Cut-off : 1
Rules in Decision : 13
Number of nodes : 11
Number of Edges : 8

KEGG Pathways

Legionellosis	2
Cytokine-cytokine receptor i...	2
Toll-like receptor signaling p...	2
NOD-like receptor signaling ...	2
Hematopoietic cell lineage	2

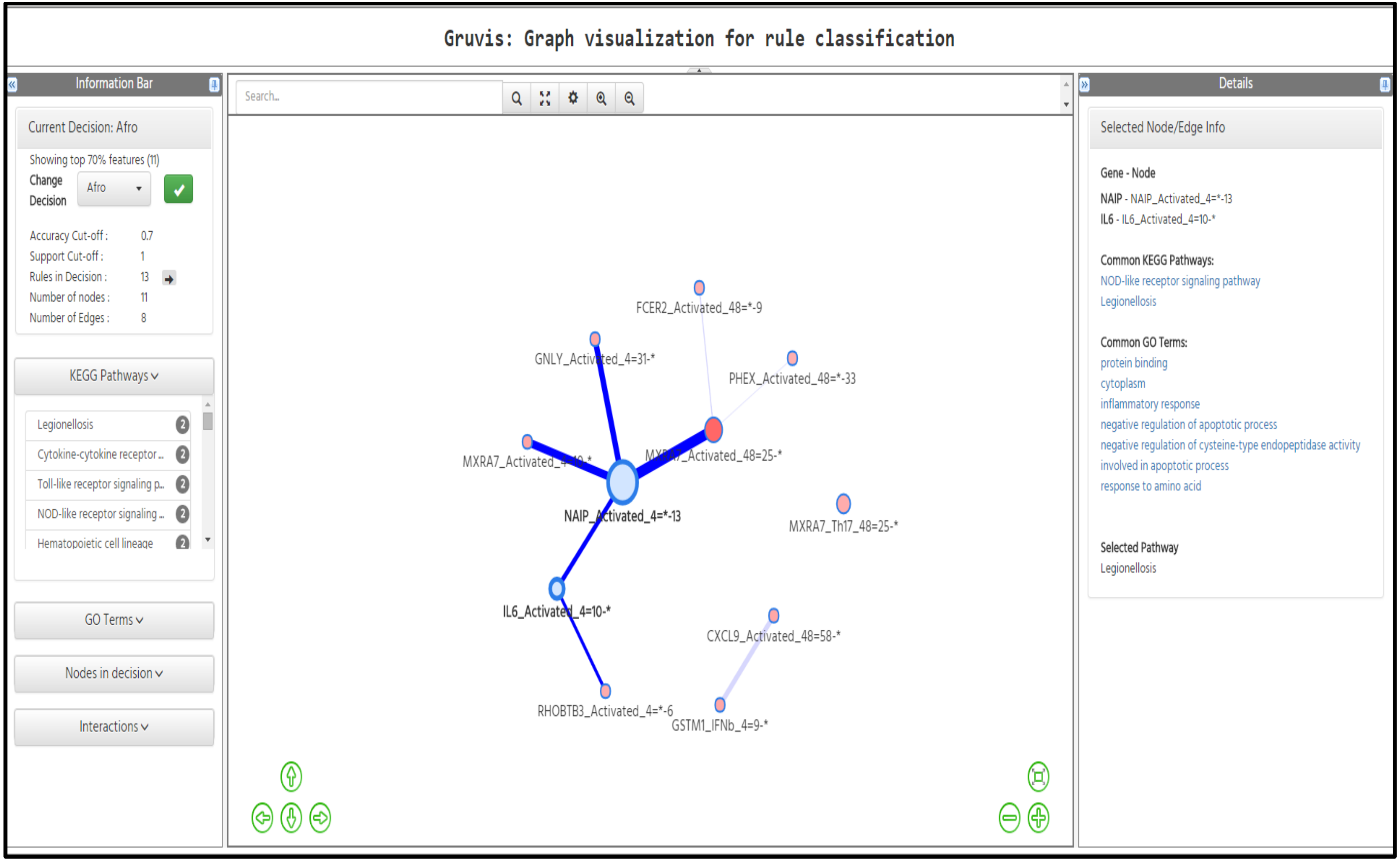
GO Terms

protein binding	5
integral component of mem...	4
external side of plasma mem...	3
inflammatory response	3
cytoplasm	3

Nodes in decision

Interactions

In addition, the details pane of a selected node will show links to the gene in question (on GeneCards website), links to the pathway diagrams (on KEGG website) and the links to the GO terms (on the AMIGO site). If more than one node is selected, the terms that are common between them as well as the common pathways are displayed (if any) as shown.



Gruvis: Graph visualization for rule classification

Information Bar

Current Decision: Afro

Showing top 70% features (11)

Change Decision: Afro

Accuracy Cut-off :	0.7
Support Cut-off :	1
Rules in Decision :	13
Number of nodes :	11
Number of Edges :	8

KEGG Pathways

Legionellosis	2
Cytokine-cytokine receptor ...	2
Toll-like receptor signaling p...	2
NOD-like receptor signaling ...	2
Hematopoietic cell lineage	2

GO Terms

Nodes in decision

Interactions

Details

Selected Node/Edge Info

Gene - Node

NAIP - NAIP_Activated_4=*-13

IL6 - IL6_Activated_4=10-*

Common KEGG Pathways:

NOD-like receptor signaling pathway

Legionellosis

Common GO Terms:

protein binding

cytoplasm

inflammatory response

negative regulation of apoptotic process

negative regulation of cysteine-type endopeptidase activity involved in apoptotic process

response to amino acid

Selected Pathway

Legionellosis