Interpretable classifiers of high-throughput cancer data using biological knowledge

Tycho Bismeijer

Masterthesis Artificial Intelligence
Supervisors:
Laura Hollink, VU University Amsterdam
Theo Knijnenburg, Netherlands Cancer Institute
Lodewyk Wessels, Netherlands Cancer Institute
Second Reader:
Annette ten Teije, VU University Amsterdam
1 Introduction

A primary goal of cancer research is to improve prognostics, diagnostics and treatment of cancer by stratification and classification of patients in prognosis and treatment response groups. Cancer is a disease commonly understood at the molecular level, and data at that resolution, high-throughput cancer data, has recently become available for thousands of cancer tumors [1, 2]. However, ultimately, we are interested in the phenotype of a tumor, such as drug response or patient survival. So, machine learning approaches employing high-throughput cancer data to predict these phenotypes have been used, for example, to predict survival based on gene expression data [3–5], or drug response of cell lines based on gene expression data [6].

As it is a major driver for the development of new treatments, another goal of cancer research is to understand cancer biology. Such understanding might be gained by providing expert biologists insight into high-throughput cancer data. Therefore, in this project, we are interested in machine learning approaches that not only provide a good prediction, but also provide insight into the data. For example, a classifier predicting drug response might explain why a drug works on certain cancers while not on others. Also, a classifier separating a good prognosis and a poor prognosis group might help to explain why certain cancers are more aggressive than others.

In order to better understand cancer biology, we want to be able to explain differences in phenotype in terms of differences at the molecular level. One way to do this is to construct a classifier separating different phenotypic classes, and inspect the relationship it places between molecular features and phenotype, that is, interpret the classifier. Unfortunately, the best performing classifiers, random forests and support vector machines, are difficult to interpret. In this project we aim to construct an interpretable classifier. An interpretable classifier could increase insight into high-throughput cancer data and its underlying biology. Insights gained from such classifiers might suggest novel hypotheses and translate into follow-up experiments. Therefore, we expect interpretable classifiers to be a useful tool in discovering new knowledge about cancer biology.

We expect a highly interpretable classifier to have the following characteristics. First, it should use a limited number of features, a human interpreting the classifier can only keep track of so many features at the same time. Second, it should pose simple relations between features: a small logic formula is far more interpretable than other non-linear formulas. Third, it should be easy to visualize: often it is easier to understand a picture than a collection of mathematical formula.
A valuable resource that could be used to build interpretable classifiers are pathway databases. Pathways, cellular signaling networks and metabolic reaction networks, are an important level of understanding between proteins at the molecular level and phenotype. Knowledge about pathways is formalized in pathway databases such as NCI-PID [7] and Reactome [8]. These pathway databases could be used to build classifiers that use pathways as an intermediary concept between molecular features and phenotype. Such a classifier would not only be understandable at two scales (molecular and phenotypical), but also at the scale of pathways, greatly improving its interpretability. Also, pathways could be used to summarize a group of molecular features, increasing the number of features that can be used in the classifier while maintaining interpretability.

In this project we will construct an interpretable classifier employing high-throughput cancer data to predict phenotype. The interpretable classifier will explain the difference between two classes. It will be interpretable across multiple biological scales, using pathways as an intermediary level of explanation. Finally, we will show how an interpretable classifier using gene-expression data to predict subtypes in breast cancer provides insight to expert biologists.
2 Cancer Biology Background

In this chapter we will introduce the biology underlying cancer and relevant to this report, introduce the classification problem we address, and describe the biological data sources we use. An overview of the biology and data involved in cancer is shown in Figure 2.1. In this figure the arrows indicate the major direction of information flow. In the following section we will introduce phenotype and pathways, followed by a description of proteins, mRNA, and DNA.

2.1 Cancer

For a cell to develop into a cancer cell it needs to acquire several characteristics, the hallmarks of cancer [9, 10]. A hallmark can be caused by the deregulation of one of several underlying pathways. A pathway is a group of related biochemical reactions that is understood to provide a specific functionality to the cell. Although all cancer cells are expected to possess all the hallmarks, the pathways that are deregulated might be very different between different types of cancer.

An important class of pathways that are often deregulated in cancer are signaling pathways. A cell continuously reacts to its environment by sens-
2.2. Predicting phenotype from molecular data

In cancer medicine, we want to know the phenotype of the tumor, which includes how aggressive the cancer is, how long the patient can be expected to survive, or what the response to treatment is. These are difficult to predict from clinical variables. RNA expression data can be obtained from tumor samples. It can be expected that RNA expression is predictive of phenotype because RNA is translated into protein, proteins determine pathway activity, and pathways give rise to phenotype, as illustrated in Figure 2.1. It has been shown, for example, that RNA expression is predictive of survival [3, 4].

Subtypes in breast cancer

In this study, the phenotype we are interested in is the intrinsic subtype of breast cancer. The five intrinsic subtypes of breast cancer were found by Perou, Sorlie and colleagues by unsupervised clustering of RNA expression profiles [12, 13]. Subtypes are understood to be different diseases, with different underlying biology. Also, the intrinsic subtypes have clinical relevance, as they are predictive of survival and drug response. Parker and colleagues trained a classifier to assign new samples to one of five intrinsic subtypes [14].
2.3 Data sources

In this section, we will first describe the gene expression data employed by our classifier as input features. Then, two sources of biological knowledge used in constructing our classifier are described: pathway databases and Gene Ontology.

**Gene Expression Data**

Gene expression data are measurements of the number of RNA molecules per gene in a sample. It is measured in tumor samples consisting of multiple cells, so represent an average over possibly very different cells. We assume that the gene expression we measure is representative for cells in the tumor. Generally, the number of samples in gene expression data is small, in the order of 100 samples. Also, it is high-dimensional, as the human genome contains tens of thousands of genes. Furthermore, the measurements of gene expression are noisy. Due to high-dimensionality and noise, predictors employing gene expression data readily overfit. That is, it is difficult, but not impossible, to build predictors that not only predict training samples correctly, but are also able to generalize and predict unseen samples correctly.

**Pathway Databases**

Pathway databases contain knowledge of pathways and their constituent reactions in a computer readable format. A simple reaction with input, output and regulatory proteins is shown in Figure 2.2, as it would appear in a pathway database. The two major classes of molecules appearing in pathway databases are proteins and small molecules. Two examples of small molecules are glucose and water. Also, pathway databases represent molecular modifications of proteins. Such modifications might activate a protein, enabling it to perform its function. The signaling pathways we saw in Section 2.1 are an example of protein modifications carrying signals from receptors to cellular processes.

The pathway databases we use are the National Cancer Institute Pathway Interaction Database (NCI-PID) [7] and Reactome [8]. We use these two because they are comprehensive, describe human biology and have clear semantics. Clear semantics was especially lacking in KEGG [15], another important and comprehensive pathway database, which we therefore did not use.

**Gene Ontology**

Gene Ontology [16] (GO), provides a standard vocabulary about cellular components, biological processes and molecular functions. We only use the part describing biological processes. A biological process term in GO can be equated with a pathway in the pathway databases described earlier. The terms are
Figure 2.2 A typical reaction in a pathway database: Pathway 1 contains one reaction, denoted by the black square (◆). The reaction has Protein A and Protein B as input, and protein complex AB as output. Protein C activates the reactions, while Protein D inhibits it. The complex AB activates Pathway 2.

hierarchically organized by ‘is a’ and ‘part of’ relationships from very specific terms at the bottom to the most general term ‘biological process’ at the top. There are 23912 terms under biological process. GO also contains the relations ‘positively regulates’, ‘negatively regulates’ and ‘regulates’ that indicate regulation of one process by another.
3 Related Work

In the first section of this chapter we will compare our method with a typical approach to predict phenotype from gene expression data. In the second section we compare our method to pathway analysis methods, which are commonly used to gain insight into high throughput cancer data at the pathway level.

3.1 Classifiers employing high-throughput cancer data to predict phenotype

In this project we build a classifier employing gene expression data to predict phenotype. Such classifiers have already been built by others [3–5]. A typical example is the approach by van ’t Veer and colleagues [4], which classifies patients into good or poor prognosis groups based on gene expression data. First, independently from the class labels, they select about 5,000 genes from the 25,000 measured, based on at least 3 of 78 tumors giving a significant measurement for that gene. Second, from these 5,000 genes, 231 genes with a suitable strong correlation (higher than 0.3 or lower than -0.3) with the class labels, are selected. Finally, using a cross-validation scheme, they find 70 genes that perform best when used as features in a nearest-mean classifier.

As the interpretable output of the nearest mean classifier as well as other approaches such as random forests is only an ordered list of genes, the interpretability of these classifiers is limited. A list of genes does not provide much insight into the biology underlying high throughput cancer data, because the relationships between genes are not shown. In comparison, our method uses knowledge from pathway databases to improve the interpretability of such relationships.

3.2 Pathway analysis

Many current approaches to provide insight into the biology underlying high-throughput cancer data infer significantly altered pathway activity from gene level measurements. In a recent review [17], a distinction is made between overrepresentation analysis (ORA) methods, functional class scoring (FCS) methods and pathway topology methods. In the rest of this chapter we summarize the ORA, FCS and pathway topology methods based on the review in [17] and detail two specific methods, GSEA [18] and PARADIGM [19], while comparing them to our method.
As we will see, all these methods generate a list pathways, which can have low interpretability. It is difficult to get insight into how genes or pathways interact to produce a phenotype. Either because the method does not consider such interactions (ORA), or because it is difficult to visualize these interactions (PARADIGM). By building a classifier, instead of looking for significantly altered pathway activity, we are able to visualize a complete overview of our analysis including interactions between genes or pathways.

**Overrepresentation Analysis**

ORA methods first generate a list of genes of interest, which often is a list of that genes are significantly differentially expressed between two phenotypes. Then, the number of genes from that list in every pathway is counted. Also, the total number of genes in a pathway is counted. A pathway can then be tested for overrepresentation of genes in the list of significantly differentially expressed genes. Chi-square, hypergeometric and binomial tests are commonly used. ORA methods result in a list of pathways with significant overrepresentation of significantly differentially expressed genes. A pathway with an overrepresentation of significantly differentially expressed genes is considered to contribute to determining the phenotype of interest. [17]

Compared to ORA, our method does not perform the first step of filtering out genes that are not significantly differentially expressed. In addition, our method allows non-linear combinations between genes, which allows our method to find informative combinations of genes which individually are not significant.

**Function Class Scoring**

In general an FCS method consists of three steps. First, a gene level statistic is computed for every gene, such as correlation between gene expression and a class label or a t-test statistic between two classes. Second, a pathway level statistic is computed per pathway using the gene level statistics. The choice of pathway level statistic is where FCS methods differ most between each other. Finally, a permutation test is done to test for the significance of the pathway level statistic. Either the class labels for each sample can be permuted, or the gene labels for each pathway can be permuted. [17]

In the evaluation we will compare our method with an FCS method: GSEA [18]. In GSEA a list of genes is ranked by the correlation with a two-valued class label, the gene level statistic. The pathway level statistic used tests for pathways with a high number of genes that are all ranked high or all ranked low in the list. Intuitively, it measures how many genes in the pathway are clustered together in the top or bottom of the ranked list. Every pathway is then tested for significance with a permutation test that permutes class labels of the samples.
Pathway Topology

Pathway topology methods are an refinement of the FCS method. They take pathway topology into account when computing pathway level statistics. [17] For example, proteins that are central to a pathway might be considered to more important than proteins in the periphery of pathway.

PARADIGM [19] is one of the more interpretable pathway topology methods. PARADIGM integrates gene expression and copy-number data in a probabilistic graphical model with a topology based on pathway structures extracted from pathway databases. The NCI-PID pathway database is translated into a graph: molecules and pathways as nodes, and regulatory relations as edges between the nodes. A node can assume three possible values, both gene expression and copy-number data is discretized into three levels: upregulated, normal and downregulated. Edges denote either a positive relationship, or a negative relationship. For example, an mRNA molecule has a positive edge to the protein it is translated into. Another example is a complex that negatively regulates a pathway, which is encoded in the graph as a negative edge between that complex and pathway. In this model, certain nodes, such as pathways, have an unspecified value. Their value is inferred using probabilistic graph algorithms. In this way, pathway level activity is inferred.

The probabilistic graph model PARADIGM uses is interpretable in principle, and the paper shows a visualization of the analysis results of one pathway. This is interpretable when looking at one pathway at a time, but, when giving an overview of the analysis, the method can only give an ordered list of pathways. Although PARADIGM represents the interactions between genes and pathways, it remains difficult to visualize this highly connected graph of probabilistic interactions in an interpretable way.
4 Interpretable Hierarchical Classifier

An overview of our approach to constructing a interpretable classifier, and datasources used, is shown in Figure 4.1. Biological knowledge bases such as GeneOntology and Reactome are used to construct a hierarchy. This hierarchy is then used to construct a classifier producing interpretable models. Finally, the classifier is visualized using web technologies.

4.1 Constructing the Hierarchy

We construct a hierarchy of at least three levels, shown in Figure 4.2, of proteins grouped in pathways. The hierarchy is organized in levels, with proteins in the lowest level, and pathways in the higher levels. We have two approaches to building a hierarchy: one using the Reactome [8] pathway database, and one using the Pathway Interaction Database by the National Cancer Institute [7] (NCI-PID) in combination with Gene Ontology [16] (GO).

Although our method could work with hierarchies with any other number of levels, we found three levels to be optimal for the following reasons. Both with using Reactome and NCI-PID, when only using two levels, we found the number of top level nodes too large: about 500. Detailed statistics on the hierarchies are shown in Section 5.1. By adding a third level the number of top level nodes can drop as low as 11, improving interpretability by reducing the number of items that need to be shown to the user at the same time. Adding another level could not reduce the number of top level nodes even further and would make the hierarchy only more complex. Therefore we construct hierarchies of three levels.

Reactome

In Reactome, pathways are groups of biochemical reactions organized into a hierarchy. At the lowest level, pathways contain only a few reactions. Higher level pathways consists of lower level pathways. Reactome provides graphical representations of mid-level pathways. In general terms, to extract a three-level hierarchy from Reactome, as shown in Figure 4.2-a, we flatten the hierarchy already available in Reactome.

More in detail, to extract a hierarchy from Reactome we start at the lowest level. For all the lowest level pathways, we include all molecules that are involved in the pathway and all the molecules that control the reactions in
Figure 4.1 Overview of our approach: A Pathway Database and Gene Expression Data are always used, while GO is only used if the pathway database is NCI-PID. Steps in italic text are described in the next chapter, other steps are described in this chapter.

the pathway. From these molecules we only retain proteins and all proteins included in complexes. In this way we group all proteins shown in the graphical representation of a pathway with that pathway. Also, all proteins in Reactome are included in at least one low level pathway.

The lowest level pathways are then grouped by the second highest level pathways. The highest level pathways, for example ‘signal transduction’ or ‘disease’, are not used because we found these not specific enough to convey useful information. Mid-level pathways could be used as an intermediary level in our hierarchy, but we do not do so to keep the hierarchy simple and interpretable. In the end, after learning a classifier, only a few pathways are kept, so a deep hierarchy is not needed to maintain an overview of all pathways.

NCI-PID
In NCI-PID, pathways are not organized in a hierarchy, so we use the pathway hierarchy from GO. Also they include more reactions per pathway than the lowest-level pathways in Reactome. Furthermore, pathways appear in two forms. First, as a group of reactions that is graphically represented and curated as a unit. We refer to these as pathway pictures. Second, as being
4.1. CONSTRUCTING THE HIERARCHY

Protein
Protein
Protein
Protein
Protein
Protein
Protein
Protein
Biological Process
Biological Process
Biological Process
Biological Process
Hallmark
Hallmark
Biological Process
Biological Process
Pathway
Pathway
Pathway
High-Level Pathway
High-Level Pathway
Pathway
Pathway
Pathway
(a) NCI-PID
(b) Reactome

Figure 4.2 Hierarchy as created from two different pathways databases: Proteins are grouped into pathways and higher level pathways.

regulated by certain molecules, for example, cell proliferation regulated by an activated protein. The regulated pathways, NCI-PID refers to them as biological processes, can not necessarily be identified with pathway pictures, as there is no explicit reference between the two, although they may have similar names. In the following paragraphs we describe how we assign proteins to biological processes that they regulate directly or indirectly, and how we group biological processes by the pathways they appear in, by GO terms, or by the hallmarks of cancer.

Grouping proteins in biological processes

All the proteins upstream of a biological process are grouped with that biological process. An example of a pathway and its associated upstream proteins is shown in Figure 4.3. A protein upstream of a biological process has a possible signaling pathway to that biological process. More precisely, a protein is upstream of a biological process when it directly regulates that process or when it is the input of or regulates a reaction producing a protein upstream of the biological process.

We found that the definition in the previous paragraph can group a lot of proteins with one biological process: 42% of the biological processes contain more than 400 proteins, as shown in Figure 4.4. Also, some proteins are grouped with a lot of biological processes: over half of the proteins are associated with more than half of the biological processes, as shown in Figure 4.4. This could cause at least two problems. First pathways show a high degree of
Figure 4.3 Upstream proteins in NCI-PID: All proteins except for H are upstream of the biological process in this picture. Protein F is upstream of the biological process because it is a constituent of the FP protein complex that is upstream the biological process.

overlap, reducing the explanatory power of one pathway being selected over another by the classifier. Second, a pathway that includes over 10% of all proteins is rather unspecific, so it is unclear what such a pathway represents. We use two approaches to reduce the number of proteins grouped with a biological process while retaining as many as possible relevant proteins. We call these two approaches explained in the following paragraphs in-pathway and no-hubs and we call the approach described in the previous paragraph unfiltered.

Before explaining the two approaches we did use we want to show that a simpler approach did not work satisfactorily. We considered directly removing biological processes containing the most proteins, or removing proteins associated with the most biological processes. In Figure 4.5, we can see that to remove proteins with a large number of biological processes we have to remove 800 proteins out of 2004. In Figure 4.4 we can see that to remove biological processes with a high number of proteins we have to remove almost half of the biological processes. Considering the large amount of discarded in this process, we decided against this approach.

For the in-pathway approach, when searching for upstream proteins of a biological process, we only consider reactions that occur in the pathway picture of that particular biological process. The NCI-PID database is curated, that is, pathways are checked by experts for correctness. For the curating process a single expert is assigned to every pathway picture. That implies
4.1. CONSTRUCTING THE HIERARCHY

Figure 4.4 Histogram of the number of proteins per biological process in the hierarchy created from NCI-PID: Although more than half of the biological processes have but a small number of associated proteins, 42% has more than 400 allocated proteins, 20% of the 2004 proteins in total.

Figure 4.5 Histogram of the number of biological processes per protein in the hierarchy created from NCI-PID: More than half of the proteins are associated with only a limited number of biological processes. The total number of biological processes is 432. Over 800 proteins are associated with more than half of the biological processes
that a pathway picture is curated as a whole, but the complete database is not curated as a whole. So, by combining all pathway pictures we can introduce contradictions and reach false conclusions. That is, a fact can be true in the context of one pathway picture and another fact true in the context of a different pathway picture, but when combined they can result in false conclusions. More specific, proteins can be upstream of a biological process even if they appear in a different pathway picture than that biological process due to molecules appearing in multiple pathway pictures, as shown in Figure 4.6. This can be prevented by staying within the pathway picture a biological originally appears in when searching for upstream proteins. That is, by only considering reactions within the pathway picture a biological process appears in, we make sure only proteins relevant to that biological processes are associated with it.

In the no-hubs approach, we remove molecules with a high betweenness centrality [20] from NCI-PID before grouping proteins with biological processes. Molecules with a high betweenness centrality could act as hubs in the network, connecting unrelated signaling networks together. For example, ATP, the major energy carrying molecule in a cell, appears in almost every pathway and could connect unrelated pathways. Betweenness centrality is computed on a graph with molecules as nodes and with directed edges originating from molecules that regulate or are input of a reaction to all molecules that are output of that reaction. Molecules with the highest betweenness centrality in this graph are removed. As the experiments in section 5.3 show, the number of nodes removed does not influence the classification performance much. Also, already in the top 20 of highly central molecules, signaling molecules associated with cancer (PIP3, GTP bound RAC1, and activated MAPK1 or MAPK3) show up. Therefore, we remove only the top 100 molecules, to minimize the number of interesting molecules being removed.

**Grouping biological processes in higher level pathways**

After proteins are grouped with biological processes, the biological processes could be grouped into higher level pathways in at least three ways. We only use the the third approach, but we will discuss two alternative approaches first. First we could group the biological processes by the pathway pictures they appear in. As the number of pathway pictures is almost as large as the number of biological processes, this approach is not very successful in providing a higher level of abstraction, and we will not consider it further. Second, we could group the biological processes by high-level GO terms, as explained in the following paragraph. The high-level GO-terms differed wildly in their specificity. Some very general (behavior), and others quite specific (establishment of cell polarity). Therefore, we have covered this approach not in as much as detail as the third approach. In the third approach we group biological processes with the hallmarks of cancer, as described in the second subsequent paragraph.
Figure 4.6 Pathway pictures in NCI-PID can overlap: Pathway pictures are shown as boxes with a dotted border. The biological process depicted here has Protein A and Protein B upstream because they occur in the protein complex AB. It has the proteins C, H and I upstream because they regulate reactions that are upstream. Also, it has Protein P and Protein Q upstream, although they are in a different pathway picture.
All biological processes in NCI-PID contain a reference to a GO term. GO contains a hierarchy of biological processes with ‘is-a’ and ‘part-of’ relationships. We use GO-terms one or two levels under the ‘biological process’ term as higher level pathways. A biological process is grouped with a higher level GO term if it is in a hierarchical relationship under that pathway.

To group biological processes with hallmarks we first associate one or more high-level GO terms with every hallmark manually. The specific association is show in Appendix A. A biological process is then grouped with a hallmark if it is in a hierarchical relationship under one of the GO-terms associated with that hallmark.

Tools used

The NCI-PID and Reactome were downloaded from their websites on 25 October 2012 in Biopax 3 format. Biopax 3 is an RDF/OWL ontology for pathway databases [21]. Both pathway databases were loaded in separate OpenRDF Sesame stores. GO was downloaded from its website in a RDF format that follows their own simple ontology and loaded in an OpenRDF Sesame store. All Sesame repositories were configured with inferencing disabled.

To compile the pathways into a hierarchy we wrote a Python package and Python scripts. The Python SPARQLwrapper 1.5.2 package was used to query the RDF repositories. The resulting hierarchies were analyzed using R 2.15. We used Gephi 0.8.1-beta to compute the betweenness centrality of molecules in the interaction network.

4.2 Hierarchical Classifier

In this section we show how we construct a classifier by combining multiple weak classifiers and using the structure of the hierarchy we constructed in the previous section. We assume a general classification task of predicting phenotype from gene expression features. In the first part of this section we will explain how to construct a hierarchical classifier, given a hierarchy as constructed in the previous section and a base classifier. The base classifier is the building block of the hierarchical classifier and should be simple as well as interpretable. The two types of base classifiers we have used, decision trees and logic formulas, will be explained in the second part of this section.

Constructing a Hierarchical Classifier

Here we will describe the process of constructing a hierarchical classifier. We use an example of a hierarchical classifier using the NCI-PID hierarchy with hallmarks. In general, any of the hierarchies constructed in the previous section can be used to construct a hierarchical classifier. Results of constructing hierarchical classifiers with different underlying hierarchies are described in Chapter 5.
4.2. HIERARCHICAL CLASSIFIER

Figure 4.7 Constructing a hierarchical classifier: In this figure a hierarchy derived from NCI-PID and with hallmarks is shown. An hierarchical classifier is constructed with a hierarchy as a template (A). The dotted lines denote the preliminary nature of the relationships and nodes in this structure. A single trained node is shown in B. As a node is trained we see how only a few of its input features are selected (solid line), while the relationships to other features are left out. As the training process proceeds, from pathways (C) to hallmarks (D), we see how the hierarchy is pruned by removing unused relationships and nodes. Finally a node with the top level classifier is added and trained (E).

The hierarchical classifier is constructed by training a base classifier at every node. In Figure 4.7, the process of constructing a hierarchical classifier is illustrated. This process starts with a hierarchy as constructed in the previous section, shown in Figure 4.7-A. In the following paragraphs we will explain how the construction process proceeds in detail.

First, a base classifier is trained at every node of the lowest, pathway, level. In Figure 4.7-B we see the results of training a base classifier at the first node of the pathway level. The input features that a base classifier at a pathway node can use are the gene-expression values of the proteins associated with this pathway in the hierarchy. The output of a base classifier is the high-level phenotype: a base classifier predicts the final class labels, no pathway activity levels are predicted. That is, one trained node is already predicting high-level phenotype from gene expression data.

All the other nodes at the pathway level are trained in the same way as described in the previous paragraph (Figure 4.7-C). Nodes are trained independently from other nodes at the same level. We can think of the group of clas-
sifiers we have trained now as an ensemble of weak classifiers [22]. Although every single classifier has only low classification performance they might be combined at the next level in a single classifier that has high classification performance. Because every pathway has a different set of associated proteins, every base classifier has different input features. So, we expect the pathways to produce classifiers covering different aspects of the classification problem, every pathway classifies a different set of samples correctly. Therefore, by combining various classifiers from the pathway level at the next level we can expect to improve classification performance beyond that of the pathway level.

Second, the nodes at the second, hallmark, level are trained, as shown in Figure 4.7-D. The output of the base classifier at a hallmark node is, once again, the high-level phenotype. The input features of the base classifier at a hallmark node are the predictions of the pathway nodes associated with that hallmark node. That is, predictions of the pathway level are used as features for the hallmark level. If using the same data to train both levels this procedure might lead to increased overfitting, because training a level can amplify the overfitting from the previous level. This increase in overfitting could be avoided by splitting the data in non-overlapping sets, and using a separate set of data to train every level. Results of classifiers with and without data splitting over levels are shown in Chapter 5.

Finally, we add a new top level with a single node to the hierarchy, as shown in Figure 4.7-E. We train a base classifier at the single node of this level which can use the predictions produced by any of the hallmarks as input features. The output of the classifier is, again, predictions of high-level phenotype. These predictions from the top level node are the predictions of the hierarchical classifier.

During the training procedure the classifier uses only a small number of the possible input features, resulting in large number of proteins, pathways and hallmark nodes not being used. These unused nodes can be left out, pruning the hierarchy. In the end, only a small number of remain in the hierarchy. This enables the visualization of the complete hierarchy, and greatly enhances the interpretability of the classifier over the hierarchy alone.

When predicting with the hierarchical classifier, first the base classifiers of the pathway level are evaluated, using gene expression features as input, resulting in an array of predictions. Using these predictions as input, the base classifiers of the hallmark level are evaluated, resulting in a smaller array of predictions. Finally, using the prediction of the hallmark level as input, the top level classifier is evaluated, resulting in a single prediction. This prediction is the prediction of the hierarchical classifier.

### Decision Trees

One of classification methods we use as a base classifier are decision trees [23]. We use decision trees because they are highly interpretable and widely used in machine learning. The decision tree classifier from the Python scikit-learn
package, which implements the popular CART algorithm for learning decision trees, is used.

A potential problem with decision trees is overfitting. When training, as a decision tree grows the classification performance on unseen data goes up at first, but when the tree grows too big classification performance goes down [23]. CART already tries to minimize this effect by pruning the tree, but this might not be completely effective. Therefore we use two other methods to prevent trees from growing too big. First, we limit the depth of decision trees to a fixed maximum. Second, we set a minimum leaf size, that is, every leaf node of the decision tree should contain a minimum number samples from the train set. The optimal settings for these two parameters will be determined in Chapter 5.

Logic Formulas

The second classification method we use a base classifier are optimal logic formulas. Logic formulas as a classifier were developed by Knijnenburg et al. [24], as an extension of Boolean function synthesis [25] with the ability to generalize over noise in the data.

This method finds formulas of Boolean logic in disjunctive normal form. The input variables of such a formula are the input features, and the value of the evaluated formula is the prediction it gives. By controlling the maximum number of disjuncts (OR-terms) and conjuncts (AND-terms) the size of the trained formula can be controlled. Using integer programming, the training procedure finds a small formula that produces the least number of errors on the train set.

In their work [24], Knijnenburg et al. show the applicability of their approach to a bioinformatics setting. They use a limited number of features (225), since the time it takes to optimize a logic formula on the expression data of all genes would be prohibitive. With the hierarchical classifier it is possible to use the logic formula classifier on gene expression data. Because a single node has only a limited number of input features, it is possible to train a logic formula at a node. The overall classifier then uses more features than a single logic formula could. The computational complexity of the hierarchical classifier as compared to training a single logic formula is reduced by disallowing direct interactions between genes in different pathways.

We have implemented a Python port of the original Matlab implementation of the logic formula classifier by Theo Knijnenburg. IBM CPLEX is used to solve the integer programs.

4.3 Visualization Design

In order for the hierarchical classifier to be interpreted it needs to be visualized. We have made two designs for the visualization, as shown in Figure 4.8 and 4.9.
A screenshot of the implemented visualization, which is based on the second design is shown in Figure 6.1. In this section we will discuss and compare the two designs as well as the design of the implemented visualization.

The first design, show in Figure 4.8, has three panels that each show a different aspect of the hierarchical classifier. The left panel shows the structure of the hierarchical classifier. In this panel, a node of the hierarchical classifier is shown as a colored disc. It is colored by the level of the node. Arrows are drawn for the input features of a node to that node. The classification performance of a node is shown as the size of a disc, making it easy to identify important nodes. When selecting a node, more details can be shown in the lower right panel. In this panel, the logical formula of a node is shown as a logical circuit, with its input features on the left and the nodes of the next level this node is connected to on the right. By clicking on these nodes the hierarchy can be navigated from node to node. All genes, pathways and hallmarks used in the visualized hierarchical classifier are shown in the upper right panel, allowing a protein, pathway or hallmark to be selected by name. The advantage of this visualization is that different information is shown in separate panels, so it is not showing too much information at the same time.

The second design, shown in Figure 4.9, shows all information in its left panel. This visualization has been designed to show hierarchical classifiers using logical formulas as a base classifier. The logical formulas are shown using the and/or graph formalism [26, pg. 108]. With this formalism, all terms of the logical formula of a node, that is, the input features of a node, are connected with that node by an edge. Edges colored red denote negation. Two edges connected by an arc denotes a conjunction between two input features, while the absence of an arc between two edges denotes a disjunction between input features. The same logical formula is shown as a logical circuit in the right panel for the selected node. The selected node is marked by a pink border. Classification performance of a node is shown as the thickness of edges going out of that node. The advantage of this visualization is that all information can be shown at the same time. Also, we found that thick edges guide the eye to important nodes, facilitating focusing on the important parts of the visualization.

We chose to use the second design for the visualization of the hierarchical classifier because it shows all the details of the hierarchical classifier at once. Because it show everything at once, the second visualization might be too big to display. If the list of genes at the left of the visualization would not fit into one screen the second visualization loses its advantage of showing the complete hierarchical classifier at once. In this case, the first visualization might preferable. But, the hierarchical classifier we build for evaluation with expert biologists in Chapter 6 has at most 64 genes, 16 pathways and 4 hallmarks to display. The second visualization is capable of displaying hierarchical classifiers of that size in one screen, so we use that visualization.

The visualization as finally implemented is shown in Figure 6.1. In comparison to the second design, we added colors and label to better distinguish
between the levels of the hierarchy. Also, we added bars to the top of nodes
to show classification performance. Finally, we removed the right panel, as
the logic circuit shows exactly the same information as the and/or graph.
When clicking on a node a popup provides reference links to Uniprot, GO or
NCI-PID.
Sustaining Proliferative Signaling

G2 Phase

Recruitment of NuMA to mitotic centrosomes

Prostanoid ligand receptors

Adrenoceptors

This Pathway in GO

This Pathway in NCI-PID

Size of a circle shows classification performance of a node.

Figure 4.8 First Design of the Visualization: The left panel shows the structure of the hierarchical classifier. The upper right panel would allow for selection of pathways by name. The lower right panel shows the logical relation between a node and right panel would allow for selection of pathways by name. The lower right panel shows the logical relation between a node and

Overview Classifier

Luminal — Basal

Overview Classifier

Node Detail

Sustaining Proliferative Signaling

Node Detail
Figure 4.9 Second Design of the Visualization: The left panel shows the hierarchical classifier. Proteins on the left and the top level classifier on the right with pathways and hallmarks in between. The logic formula of a node is shown using the and/or graph formalism, with a red edge denoting a negation. Edge thickness denotes classification performance of the node at left side of the edge. The right panel shows a single node in more detail, with the logical formula shown as logic circuit and reference links to biological knowledge bases.
5 Results

In this chapter we show the technical results of using the hierarchical classifier. We show results for four different approaches to constructing the hierarchy underlying a hierarchical classifier. The first three, unfiltered, no-hubs and in-pathway, derive a hierarchy from NCI-PID, as explained in Section 4.1. The fourth derives a hierarchy from Reactome. In the first section of this chapter we compare the hierarchies while not yet using a hierarchical classifier. The second section introduces the dataset used in this and the next chapter. Finally, we assess of classification performance of the hierarchical classifier in the third section of this chapter.

5.1 Comparing Hierarchies

In this section we compare the hierarchies we have constructed from NCI-PID using the unfiltered, no-hubs and in-pathway approaches and the hierarchy constructed using Reactome. For the no-hubs approach we show results for removing the top 100, 500 or 1000 hubs. The hierarchies are compared on the number of proteins in pathways and compared on the number of pathways associated with proteins.

The number of proteins contained in a pathway should be low so pathways are different from each other and highly specific. When training the classifier we select only a few proteins from pathway, so if pathways contain a large proportion of all proteins we might select the same proteins for a large number of pathways. Also, the name we give to a pathway might not specifically apply to the few proteins we select out of it. We want the number of pathways associated with a pathway to be low for similar reasons. If a protein is associated with a large number of pathways, the pathways become similar to each other and lose specificity.

A frequency table of pathways for different numbers of contained proteins is shown in Table 5.1. A frequency table of proteins for different numbers of associated pathways is shown in Table 5.2. In both tables we see that removing a larger number of hubs leads to larger reductions in pathway and protein overlap, and that removing 1000 hubs is as effective as the in-pathway approach. However, the in-pathway approach leaves pathways out of the hierarchy while the no-hubs approach does not and the in-pathway approach leaves a larger number of proteins out of the hierarchy than the no-hubs approach. Pathways are left out of the hierarchy in the in-pathway approach because certain biological processes in NCI-PID do not have upstream proteins within their pathway picture. Additionally, the hierarchy from Reactome contains more proteins and pathways than NCI-PID.
5.1. COMPARING HIERARCHIES

### Pathway frequency in a Hierarchy

<table>
<thead>
<tr>
<th>Number of Proteins</th>
<th>Unfiltered</th>
<th>No Hubs (100)</th>
<th>No Hubs (500)</th>
<th>No Hubs (1000)</th>
<th>In Pathway</th>
<th>Reactome</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–10</td>
<td>305</td>
<td>343</td>
<td>423</td>
<td>456</td>
<td>258</td>
<td>518</td>
</tr>
<tr>
<td>10–100</td>
<td>40</td>
<td>73</td>
<td>140</td>
<td>130</td>
<td>171</td>
<td>269</td>
</tr>
<tr>
<td>100–500</td>
<td>40</td>
<td>156</td>
<td>33</td>
<td>10</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>500–1000</td>
<td>191</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000+</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>596</td>
<td>596</td>
<td>596</td>
<td>596</td>
<td>439</td>
<td>809</td>
</tr>
</tbody>
</table>

**Table 5.1 Frequency table of pathways for ranges of number of contained proteins**

### Protein frequency in a Hierarchy

<table>
<thead>
<tr>
<th>Number of Pathways</th>
<th>Unfiltered</th>
<th>No Hubs (100)</th>
<th>No Hubs (500)</th>
<th>No Hubs (1000)</th>
<th>In Pathway</th>
<th>Reactome</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–10</td>
<td>1337</td>
<td>1399</td>
<td>1618</td>
<td>1682</td>
<td>1575</td>
<td>4036</td>
</tr>
<tr>
<td>10–25</td>
<td>50</td>
<td>138</td>
<td>199</td>
<td>136</td>
<td>143</td>
<td>246</td>
</tr>
<tr>
<td>25–50</td>
<td>3</td>
<td>50</td>
<td>105</td>
<td>5</td>
<td>20</td>
<td>7</td>
</tr>
<tr>
<td>50–100</td>
<td>267</td>
<td>289</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>100–200</td>
<td>0</td>
<td>389</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200+</td>
<td>744</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>2401</td>
<td>2266</td>
<td>1928</td>
<td>1823</td>
<td>1740</td>
<td>4291</td>
</tr>
</tbody>
</table>

**Table 5.2 Frequency table of proteins for ranges of number of associated pathways**

One level higher in the hierarchy than pathways we have placed hallmarks for NCI-PID, or high level pathway for Reactome. In Table 5.3 we show a frequency table of hallmarks and high level pathways for different numbers of contained proteins, and in Table 5.4 we show a frequency table of proteins for different number of associated hallmarks or high level pathways. We see the same pattern as in the two tables discussed in the previous paragraph. For the no-hubs approach a large number of hubs needs to be removed to get an hierarchy as simple as from the in-pathway approach, but the in-pathway approach leaves a large number of proteins out. Additionally, the hierarchy derived from Reactome is larger than the hierarchies derived from NCI-PID.
### Table 5.3 Frequency table of hallmarks and high level pathways for ranges of number of contained proteins:

For the hierarchies derived from NCI-PID the number of hallmarks is given, for the hierarchy derived from Reactome the number of high-level pathways is given.

<table>
<thead>
<tr>
<th>Number of Proteins</th>
<th>Hallmark frequency in a NCI-PID Hierarchy</th>
<th>High-Level Pathway Frequency in Reactome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unfiltered No Hubs</td>
<td>No Hubs (100)</td>
</tr>
<tr>
<td>0–10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10–50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50–100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100–500</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>500–1000</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>1000–2000</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table 5.4 Frequency table of proteins for ranges of number of associated hallmarks or high level pathways:

For the hierarchies derived from NCI-PID the number of hallmarks is given, for the hierarchy derived from Reactome the number of high-level pathways is given.

<table>
<thead>
<tr>
<th>Number of Hallmarks or High Level Pathways</th>
<th>Protein frequency in a Hierarchy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unfiltered No Hubs</td>
<td>No Hubs (100)</td>
</tr>
<tr>
<td>1</td>
<td>592</td>
<td>646</td>
</tr>
<tr>
<td>2–5</td>
<td>304</td>
<td>409</td>
</tr>
<tr>
<td>5–10</td>
<td>881</td>
<td>783</td>
</tr>
<tr>
<td>10+</td>
<td>153</td>
<td>19</td>
</tr>
<tr>
<td>Total</td>
<td>1930</td>
<td>1857</td>
</tr>
</tbody>
</table>
5.2 Data Set

We use the breast cancer dataset from the The Cancer Genome Atlas (TCGA) [1]. In the TCGA breast cancer paper [1] gene expression from microarrays is used. We use gene expression data from RNA-seq, which was later published by TCGA [27]. The data was downloaded in July 2012. The dataset has 779 tissue samples with the expression of 15399 genes.

More in detail, we used level 3 expression data per gene from the RNA-seq version 2 analysis protocol [27]. In this protocol, the raw data comes from Illumina HiSeq RNA-sequencing, which is processed by Mapsplice [28] and RSEM [29] to get a read count per gene. The read count per gene is then normalized by the upper quartile method [30] to get a quantification of gene expression comparable between samples. The data we downloaded was already processed in this way.

Discretizing Gene expression

The logic formulas, as described in Section 4.2, require binary input features, so we discretize the gene expression data. For every gene we try to find an expression value threshold that divides the two classes as good as possible. We set the threshold of a gene at the mean of the median expression of the samples in the first class and the median of samples in the second class. For a sample, for every gene, if the gene expression is above the gene’s threshold, the discretized gene expression is 1, otherwise it is 0.

Spurious Data Samples

After the classification performance experiments (Section 5.3) and evaluation with biologists (Chapter 6) were finished, we discovered 26 samples in our dataset had expression values very different from the other samples, as shown in table 5.5. For these spurious samples, the data we used has different expression values in comparison to data released on 17 December 2012, while the expression values were the same for the other samples. Therefore, these

<table>
<thead>
<tr>
<th>Statistic</th>
<th>26 Spurious Samples</th>
<th>753 Other Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>25.80</td>
<td>1240</td>
</tr>
<tr>
<td>minimum</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25% quantile</td>
<td>1.945</td>
<td>104.8</td>
</tr>
<tr>
<td>median</td>
<td>6.633</td>
<td>427.9</td>
</tr>
<tr>
<td>75% quantile</td>
<td>17.49</td>
<td>1555</td>
</tr>
<tr>
<td>maximum</td>
<td>19613</td>
<td>2040897</td>
</tr>
</tbody>
</table>

Table 5.5 Range of gene expression: Summary statistics of gene expression values for the spurious samples and all other other samples.
samples might have incorrect gene expression values and the data might has been corrected after we first downloaded it. To assess the impact of the spurious samples on our results the classifier used in the evaluation with biologists was constructed again, but leaving the spurious samples out of the data. Its accuracy goes up from 0.92 to 0.94. The original visualization is shown in Figure 6.1, the visualization of the hierarchical classifier constructed from the data without spurious samples is shown in Figure 6.2.

Defining Subtype

In this project we train predictors of intrinsic subtype in breast cancer. The TCGA breast cancer paper [1] gives subtypes for the samples analyzed in that paper, but we have expression data on 276 additional samples, for which no subtype data is available. The intrinsic subtypes are defined on gene expression data, which we do have, so we used the R package genefu 1.8.0 to identify subtypes from the RNA-seq gene expression data.

TCGA uses two technologies to measure gene expression: RNA-seq and microarrays. RNA-seq data represents the amount of RNA in the sample, while microarray data is the log ratio of the amount of RNA in the sample and the amount of RNA in a standard comparison solution. While we used RNA-seq data, the genefu package expects microarray data, but, as shown in the next paragraph, we got good results with log2 transforming the expression values and using genefu's robust scaling method. The log2 transform gives the RNA-seq data the same log transform inherent in microarray data, and the robust scaling centers and scales the RNA-seq data to fall in the same range as the microarray data.

To verify the identified subtypes we compare them to the subtypes given in the TCGA breast cancer paper. The subtypes given in the TCGA breast cancer paper are identified from microarray data, while we used RNA-seq, so we cannot expect complete agreement. Still, we identified the same subtype in 87% of the samples. If we remove the spurious samples, this improves to 90%. Confusion matrices are shown in Table 5.6, for all data, and in Table 5.7 for the data without the spurious samples. In both tables, we see that most errors are made between the Luminal A and Luminal B subtypes.

The classification task we use in this and the next chapter is to classify between samples with the Luminal or Basal subtype. The Luminal subtype is either Luminal A or Luminal B. Samples with one of the other subtypes are not used. In the dataset are 142 Basal samples (0.21%) and 534 Luminal samples (0.79%).

In this classification task, the class labels are defined on the features. So, in principle, it should be an easy task. Indeed, a random forest model achieves 100% accuracy. But, in this project we want to build an interpretable classifier. As applied to this classification task, the hierarchical classifier could explain the difference between the luminal and basal subtypes. This will be evaluated in the next chapter, but first we assess the classification performance of the hierarchical classifier.
5.3. **CLASSIFICATION PERFORMANCE**

<table>
<thead>
<tr>
<th>Genefu RNA-seq</th>
<th>TCGA Paper</th>
<th>Basal</th>
<th>Her2</th>
<th>LumA</th>
<th>LumB</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basal</td>
<td>89</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Her2</td>
<td>1</td>
<td>48</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>LumA</td>
<td>2</td>
<td>5</td>
<td>180</td>
<td>22</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>LumB</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>99</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6 Confusion matrix of subtypes from TCGA paper and Genefu RNA-seq: Genefu was run on the RNA-seq data including the spurious samples.

<table>
<thead>
<tr>
<th>Genefu RNA-seq</th>
<th>TCGA Paper</th>
<th>Basal</th>
<th>Her2</th>
<th>LumA</th>
<th>LumB</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basal</td>
<td>89</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Her2</td>
<td>1</td>
<td>48</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LumA</td>
<td>2</td>
<td>2</td>
<td>192</td>
<td>14</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>LumB</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>94</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.7 Confusion matrix of subtypes from TCGA paper and Genefu RNA-seq: Genefu was run on the RNA-seq data excluding the spurious samples.

### 5.3 Classification Performance

In this section we measure the accuracy of the hierarchical classifier on the classification task defined in the previous section for different parameter settings. All these measurements were done in a cross-validation scheme. We used 50 folds when using trees as a base classifier, or 10 folds when using logic formulas as a base classifier. As the discretization of the data, explained in the previous section, already learns from the class labels, discretization is done inside a fold. This is done by finding the per gene thresholds on a train set, and discretize both the train and test set using these thresholds.

Because decision trees are well described in the literature, and are easy to manipulate, we do most experiments in this chapter with trees as a base classifier. For the evaluation with biologists we have chosen logic formulas over trees because the visualization we wanted to use is designed with them in mind. Therefore we also asses the performance of hierarchical classifiers with logic formulas as a base classifier.
CHAPTER 5. RESULTS

Limiting Tree Size

As explained in Section 4.2, we limited tree size by setting a minimum leaf size or by setting a maximum tree depth. Classification performance for different settings of minimum leaf size are shown in Figure 5.1-a. As the minimum leaf size increases and the tree is becoming smaller, accuracy goes down. Classification performance for different settings of maximum tree depth is shown in Figure 5.1-b. Setting maximum tree depth lower than 3 hurts performance, while there is no indication that setting any maximum tree depth increase performance. Both plots show no evidence of overfitting. If any overfitting is happening in the trees, setting a minimum leaf size or maximum tree depth is hurting performance more by limiting the trees in size than improving performance by reducing overfitting.

![Figure 5.1 Classification performance when limiting tree size: Accuracy when setting a minimum leaf size or maximum tree depth for different underlying hierarchies. For these two graphs we used separate data per level and discretized expression data.](image)

Separate data per layer

In Section 4.2 we argued using separate data per level when training the hierarchical classifier is required to prevent overfitting. To verify this we repeated the experiments resulting in Figure 5.1 but training classifiers using the same data for all levels instead of separate data per level.

In Figure 5.2-a the accuracy when increasing minimum leaf size is shown. Accuracy does not go down as fast with increasing minimum leaf size as in
5.3. **CLASSIFICATION PERFORMANCE**

Figure 5.1-a because not using separate data per level allows for larger trees with the same minimum leaf size: when using the same data for all levels a larger number of samples is available for training a node than when splitting the data up in separate sets. More importantly, we see that large trees, those with a small minimum leaf size, have a lower accuracy than smaller trees. A drop in accuracy when building larger trees is also shown when limiting tree depth (Figure 5.2-b). Also, for both plots, in comparison to using separate data per level, the highest accuracy remains about 0.93. These results indicate that the hierarchical classifier overfits when not using separate data per layer, but the overfitting but can be prevented by limiting the base classifiers in size.

![Figure 5.2](image)

**Figure 5.2 Classification performance when not using separate data per level:** Accuracy when setting a minimum leaf size (a) or maximum tree depth (b) for different underlying hierarchies. For these two graphs we used the same data for all levels and discretized expression data.

**Discretizing gene expression**

As explained in Section 5.2, we discretize expression data to allow using logic formulas as a base classifier. When using trees it is not necessary to discretize the data, so in this paragraph we compare performance results for discretizing data, shown in Figure 5.3, with not discretizing data, shown in Figure 5.1. Overall accuracy is higher when not discretizing. Also, the highest accuracy when not discretizing is higher (0.95) than when not discretizing (0.93).

**Removing Hubs**

When removing hubs from the network (Section 4.1) we need to determine the number of hubs to remove. In Figure 5.4 we see the accuracy of the
CHAPTER 5. RESULTS

Figure 5.3 Classification performance when not discretizing data: Accuracy when setting a minimum leaf size or maximum tree depth for different underlying hierarchies. For these two graphs we used separate data per level and did not discretize expression data.

Hierarchical classifier using hierarchies derived from NCI-PID with a different number of hubs removed. No clear trend can be seen, so removing hubs does not have a large impact on classification performance.

The lack of impact of removing hubs could be explained by the relatively simplicity of the classification task we use. A lot of genes are associated with the intrinsic subtypes of breast cancer, Parker and colleagues [14] found 1,906 different genes reported in at least one of the four studies they surveyed. So, removing a few genes does not hurt performance as there are plenty of other genes left that are also predictive of subtype.

Logic Formula Size

When training a logic formulas we set a maximum number of disjuncts and conjunctions to limit the formula in size. A size limit is required to keep the training computationally feasible and prevent overfitting. Here, we want to assess the impact of logic formula size on classification performance. As 42% of the pathways in the unfiltered NCI-PID hierarchy are associated with 400 or more proteins, we found it computationally infeasible to construct a hierarchical classifier based on that hierarchy. For the other hierarchies, we show accuracy of the hierarchical classifier with logic formulas of different sizes in Figure 5.5. We see that the most complex formulas considered, those with 2 disjuncts and 2 conjunctions, perform best.
5.3. **CLASSIFICATION PERFORMANCE**

**Figure 5.4 Classification performance when removing hubs:** Accuracy when removing a number of hubs from NCI-PID before constructing the hierarchy for trees with a maximum depth of 3, 5 or 10. For this graph we used separate data per level and discretized expression data.

**Figure 5.5 Classification performance of logic formulas:** Accuracy when limiting formula size by number of disjuncts and conjuncts for different underlying hierarchies. For this graph we used separate data per level and discretized expression data.
In this chapter, we will assess the interpretability by expert biologists of the hierarchical classifier we built. We will evaluate three aspects of interpretability. First, and foremost, we want to compare our visualization with state-of-the-art methods employing gene and pathway lists on the insight provided to expert biologists. Second, we want to assess the usability of our visualization tool. Third, the concepts we use in the classifier, such as pathways, should correspond with the concept a biologist understands in the visualization. In summary, the goal of the evaluation is to assess gained insight and check for conceptual concordance as well as usability. In the following paragraphs we explain in more detail what is meant by these three goals. Then we will explain the methods used. Finally we will present the results of this evaluation.

Our primary reason for building an interpretable classifier is that it should give rise to new insights. Though often used in the visual analytics literature, the term insight is not always very well defined. In the following sentences we summarize a definition of insight in visual analytics by Chang et al. [31], give by showing how it used, and by contrasting it with the definition of insight in cognitive science. Insight is a substance, as a word it is used in the same manner as knowledge or information. It can be gained, discovered or provided. By contrast, in cognitive science, insight occurs as an ‘aha!’ or Eureka-moment. It can be experienced or had, and is momentary. Here, we follow the definition Saraiya et al. [32] use: “a unit of knowledge gained from the analysis of data”.

Prerequisite for interpretability is the usability of our visualization. If a biologist has difficulties to use the visualization, it is unlikely it will be interpreted correctly. We have designed our visualization with visual analytics and usability guidelines in mind, and the visualization is not very complex. Also the improved interpretability over existing methods should come from the new underlying hierarchical model, not from improved usability. Nonetheless, we want to identify any major usability obstructions, as they might determine the results of experiments evaluating insight.

Another prerequisite for interpretability is conceptual concordance between the visualization and the mental model of the biologist. We expect the biologist to think in terms of pathways, so we also use pathways in our model. Furthermore we assume that proteins, pathways and hallmarks are hierarchically related. We will check these assumptions. Also, we want to see how our visualization succeeds in communicating in these terms. We need to
show that a biologist can understand a pathway in our visualization as the conceptual pathway he or she already knows.

6.1 Methods

We have adapted the insight-based method by Saraiya et al. [32] to evaluate our visualization. In the following section we will first review three related methods. Then, we will introduce the insight-based method, and explain our adaption of it. Finally we discuss the research design using this method, and allowing for checking usability and conceptual concordance.

Related Methods

In a benchmark task [33] a limited task is given to a large number of users, Torry and Möller [34] recommend at least thirty participants. With this approach usability problems can be found by analyzing how users use the application, and by interviewing them after performing the task. Visualizations can be compared by measuring simple benchmarks such as time taken to solve a problem or by performance measures of a task. The task given should be very limited in scope if the results between users are to be quantitatively compared using the benchmark. Therefore benchmark tests can not be used to evaluate more elaborate tasks with a large scope, higher difficulty or tasks requiring a deeper understanding of the task involved. But insight is difficult to quantify. In contrast to benchmark tasks, the insight-based method does not require a quantitative measurement, avoiding the difficulty of quantifying insight.

A second method is doing expert reviews [34]. In an expert review a human-computer interaction expert reviews the visualization. Although this method requires a thorough description of typical users and the tasks they might perform it is unnecessary to confine the evaluation to limited tasks. We could evaluate the visualization on how much insight it gives. Torry and Möller [34] have applied this in the evaluation of visualization in the medical domain, and found that the evaluators should not only be knowledgeable about human-computer interaction, but also about visualization. In our case that would mean we need an evaluator knowledgeable about visualizations giving insight, but no such usability experts were available. Furthermore, expert reviews are good for evaluating early prototypes, but a final evaluation with users remains necessary, as expert evaluators might be wrong. The insight-based method does not require the availability of human-computer interaction experts, and does the evaluation with users.

In the insight-based method [32], users are given a task, as in benchmark tests, but performance is not measured along a benchmark. They use an open task, asking the user to analyze until he or she feels that no more insight can be gained, instead of a limited task. The method gives both qualitative and quantitative results and focuses on measuring insight. Therefore we use this
method for the evaluation. Furthermore, although enough participants were available, we had a limited amount of time available in this project for the evaluation, and only a few participants were interviewed. As we will explain in the next section, results can be achieved with a low number of participants using this method.

Insight-based Evaluation

We have adapted the insight-based method by Purvi Saraiya et al. [32]. They do a comparative evaluation of five RNA expression microarray visualization tools. An expert biologist is given a visualization tool and dataset and asked to analyze the data until they felt they could not get any more insight out of the data. The expert biologist is asked to think aloud during the analysis. They used a 3x5 between-subject design with 30 subjects in total. That is, three datasets and five visualization tools are compared, with two subjects assigned to each combination of tool and dataset. In order to measure the insight a tool gives it needs to be clearly defined. For the purpose of the evaluation insight is defined as: “a unit of knowledge gained from the analysis of data”. Using the data generated by the think-aloud protocol, distinct insights are coded along eight qualitative or quantitative characteristics. The performance of the different tools disregarding dataset is then compared along the eight defined characteristics of insight, and, if possible, related to differences between datasets.

In this investigation, we largely followed the insight-based method summarized in the last paragraph. Due to a limited duration of the project we were able to perform and analyze only four interviews. As this number is much smaller than the number of participants in Saraiya et al.’s investigation (30), we are less powered to do a quantitative comparison. Another difference is that we compared the visualizations within participants, instead of between participants. This increases power because a larger number of participants is available per visualization, and it diminishes the effects of pre-existing differences between participants. Overall we still expect to find qualitative results. A threat to validity in a within-subject design is order effects: fatigue or learning from the first visualization can influence the second. By comparing the two groups with visualizations presented in different orders we are able to estimate the magnitude of any order effects. Also, the limited number of participants allows us to evaluate the visualization on only one dataset.

Research Design

The evaluation with one of the expert-biologists consisted of an experiment and an interview. An audio recording and a screen capture was done in order to allow later analysis. For the experimental part we used the adapted insight-based method introduced in the previous section. In addition to the insight-based evaluation we also asked the expert biologists to review the
6.1. METHODS

In the evaluation we first asked the participant a few background questions, which can be found in Appendix B. We used a non-structured standardized interview [35]. That means we ask all participants the same questions, but the order might differ. This allows for a natural conversation, and is better suited to unexpected discovery than more structured interviews.

Second we trained the users in using both visualizations using example data sets. Our visualization was used to classify between cancer and normal, non-cancer, samples. In the other visualization, GSEA was used to show the differential expression between samples with the HER2 subtype and all other samples. The visualizations were explained, and participants were encouraged to ask questions until they felt they understood the visualization.

Third, we introduced the task. We introduced the TCGA breast cancer dataset. We did not introduce the expert biologists to the subtypes in breast cancer, as to not influence their analysis.

Fourth we asked them to perform the analysis using the two visualization tools. We use a within-subject design comparing two treatments: our visualization and a list of genes. We compare two groups: one group that first uses our visualization and the list of the genes generated by GSEA, and another group the other way around. In those two groups we measure the insight gained from both treatments, as further detailed in the next paragraph. Participants were asked to think aloud during the analysis, and we encouraged them to continue to do so if they were silent for about ten seconds.

Audio recordings of the analysis by participants were used to report all insights a participant had. The insights were coded using three characteristics. This is a reduced set from the method by Saraiya and colleagues.

- The insight itself. This can be used to count the number of insights, and to compare between participants or tools as to they give the same or different insights.
- Time to Insight. Time from the beginning of the analysis with the tool to the moment the participant does the observation.
- Domain Value. Coded on a scale from 1 (a trivial insight about the expression of a single gene) to 5 (a novel hypothesis that could be validated by follow-up experiments). In contrast to Saraiya et al. we coded this ourselves, and do not employ a domain expert to do so.

Also, we used the audio recordings and captured videos of the computer screen to identify major usability problems. If the visualization has major usability problems, we can expect a user complains about it or to see unexpected interaction patterns between a user and the visualization.

The fifth step is the interview. As with the questions at the beginning of the evaluation this is a non-structured standardized interview. The questions
can be found in Appendix B. In the first phase we asked about missing features and possible improvements for our visualization to assess the usability of our visualization and to identify possible improvements. In the second phase we asked questions to assess the conceptual concordance between the visualization and the biologist experts.

6.2 Results

In this section we first discuss the results of counting the insights each biologist had with GSEA and the hierarchical classifier. For the evaluation we trained a hierarchical classifier using logic formulas as a base classifier and using the hierarchy constructed with the in-pathway approach. The classification task is to distinguish between the luminal and basal subtypes, as described in the previous chapter. The visualization of the hierarchical classifier is shown in Figure 6.1. Finally, we discuss three possible improvements of the hierarchical classifier we gained from the interviews.

Counting Insights

In Table 6.1 we show the total number of insights and sum of domain values per method and expert biologist. For Biologist 1 we see the hierarchical classifier gives slightly more insights with a higher domain value. This biologist made more observations about pathways with the hierarchical classifier while more observation about genes with GSEA, which might be explained by this subject seeing pathways as gene groups. Biologist 2 made one trivial observation about a gene in GSEA, but did not made any more observations after that, despite our best efforts to encourage this subject to do so. Nonetheless this subject did make some of the more useful comments we used in the rest of this section. Biologist 3 made more observations with GSEA than with the interpretable classifier. This difference reflects this subject focusing on the hallmarks in the interpretable classifier, while focusing on finding known and new genes with GSEA, and there are more genes than hallmarks. Biologist 4 had more insights with a higher domain value with the hierarchical classifier than with GSEA. This subject recognized the hallmarks presented in the hierarchical classifier and went to pathway and gene level observations from there. With GSEA this subject only found some known genes and pathway without relating them to each other, and discarded unknown genes immediately as irrelevant.

Different Interpretations of Pathways

We found the concept of pathways is interpreted in some very different ways. We can distinguish the interpretations by the four biologists we interviewed into two groups. First, Biologist 2 and 3 mainly see pathways as a group of related genes. When using the visualization, both looked at genes first and
6.2. RESULTS

<table>
<thead>
<tr>
<th>Biologist</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insight Count IHC</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Insight Count GSEA</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Sum Domain Value IHC</td>
<td>12</td>
<td>0</td>
<td>10</td>
<td>22</td>
<td>11</td>
</tr>
<tr>
<td>Sum Domain Value GSEA</td>
<td>9</td>
<td>1</td>
<td>10</td>
<td>22</td>
<td>5.5</td>
</tr>
<tr>
<td>Avg Domain Value IHC</td>
<td>2</td>
<td>2.5</td>
<td>2.2</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>Avg Domain Value GSEA</td>
<td>1.8</td>
<td>1</td>
<td>2</td>
<td>2.4</td>
<td>1.8</td>
</tr>
<tr>
<td>Experiment Order</td>
<td>IHC-GSEA</td>
<td>GSEA-IHC</td>
<td>GSEA-IHC</td>
<td>IHC-GSEA</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1 Insight counts and domain value: for the interpretable hierarchical classifier (IHC) and Gene Set Enrichment Analysis (GSEA)

pathways later. One of these subjects stated he looked at genes first because he wanted to identify genes that he knows are indicative of activated pathways when deferentially expressed. Second, Biologist 1 and 4 think of pathways at a more abstract level, seeing them mainly as a mechanistic concept. That is, they see pathways as biological processes in a cell. Biologists 1 stated that he sees a pathway as a biologic cascade that leads to phenotypic differences in cell biology, and wanted to use the visualization to visualize dominant cell-biological mechanisms. Biologist 4 sees cancer as deregulation of pathways, and prefers to think in terms of pathways instead of genes. Both Biologist 1 and 4 looked at pathways and hallmarks first and genes later.

The hierarchical classifier allows both interpretations of pathways. In the hierarchy, pathways are groups of genes, so can be interpreted as such. For example, in the visualization used in the evaluation we see a pathway denoted as ‘Fever’, but the associated genes are also related to other pathways, so the name ‘Fever’ is not fully descriptive of the function of the associated genes. But, except for the ‘Fever’ pathway, all biologists found pathways and hallmarks to be a good summary of the underlying genes. Also, all biologists found the the three most prominent hallmarks in the visualization (Sustaining Proliferative Signalling, Resisting Cell Death and Evading Growth Suppressors) are in agreement with what they already know about the difference between basal and luminal tumors. So, the pathways can also be interpreted as biological processes.

Lack of direction in pathway activation

Biologically, pathways can be up or down regulated in one class as compared to another class. For example, a pathway can be more active in the luminal samples than in the basal samples, so it is upregulated in luminal samples as compared to basal samples. This does not necessarily mean the genes in that pathway are also highly expressed. Some genes are negative regulators of the
CHAPTER 6. EVALUATION WITH EXPERT BIOLOGISTS

pathway, and are expected to have low expression if the pathway is activated. Other genes are translated into proteins that need to be activated before they regulate a pathway, so their high expression does always result in pathway activation. As we take pathways as groups of genes and do not take the direction of regulatory relations into account, the hierarchical classifier cannot make the distinction between up and down regulated pathways. Instead it identifies deregulated pathways. However, the visualization wrongly suggests otherwise because a green line from a gene to a pathway could easily be interpreted as indicating that the gene positively regulates the pathway, while it indicates that the gene is highly expressed in luminal samples as compared to basal samples.

Two hallmarks in the visualization, Sustaining Proliferative Signaling and Evading Growth Suppressors, are similar and mainly differ in direction. Roughly speaking, the first is a negative regulation of cell growth while the second is a positive regulation of cell growth. Combined with the lack of clarity about up or down regulated pathway, as discussed in the previous paragraph, this leads to confusion. This could be the reason why Biologist 2 had almost no insights into the data. Another biologist stated that the pathway Negative Regulation of Cell Proliferation could also be grouped under Proliferative Signalling instead of Evading Growth Suppressors.

More genes

All biologists found that too few genes were displayed, and wanted to see data on a larger number of genes. One of the biologists stated that he would have liked to see more proteins that are contained in a pathway and wanted to see how the other genes in a pathway correlate with each other and the phenotype. This would allow this subject to see whether the whole pathway is deregulated, or just a few proteins are differentially expressed.
AND/OR Graph: Edges are drawn from antecedents to consequents. An arc between edges denotes an and-relationship between antecedents; no arc means or. Edge weight shows classification performance of left node. The bars on top of a node also show classification performance.

Figure 6.1
7 Discussion and Conclusion

In this project, we have constructed an interpretable classifier that explains the subtypes of breast cancer. This classifier is based upon a hierarchy constructed from biological knowledge bases. Furthermore, it is interpretable while maintaining an acceptable accuracy of 0.92. Finally, it might give more insight than current methods.

As an preliminary step to training the interpretable hierarchical classifier, we have constructed a hierarchy of proteins in pathways from biological knowledge bases. This hierarchy can function as a scaffold for the interpretable classifier. Also, the intermediate step of building a hierarchy allows different knowledge bases to be used with the interpretable hierarchical classifier, in this project we have used Reactome and a combination of NCI-PID and GeneOntology.

We have introduced an approach to building an interpretable hierarchical classifier. On the classification task we used, the accuracy of the interpretable hierarchical classifier (between 0.92 and 0.94) is not as high as current methods, such as random forests (0.99). Nevertheless, we judge this accuracy to be high enough to allow interpretation of the hierarchical classifier. Interpretability of the hierarchical classifier was shown using an adaptation of the insight-based method. All participants were able to interpret the visualization of the hierarchical classifier. Also, all but one participant found insights into the data using the visualization. In comparison to a current method, GSEA, the interpretable hierarchical classifier affords more insights on the pathway level. Therefore, we can conclude that visualization of the hierarchical classifier succeeds in being interpretable. In conclusion, the interpretable hierarchical classifier is a viable method to construct an interpretable classifier employing high throughput cancer data to predict phenotype.

7.1 Further Work

In this section we suggest some possible improvements to our method of constructing the interpretable hierarchical classifier. Also, we discuss some of the questions raised by our results.

First, it should be possible to get a better understanding into the structure of the hierarchy in connection with the interpretable classifier. For example, one pathway might be more likely to be selected than others, so we want to know what pathways a biased towards being selected by the interpretable hierarchical classifier. More generally, we might want to say something about the statistical significance of the pathways selected. These questions might be
answered by doing a permutation test with permuting either class labels or gene labels.

**Improving Classification Performance**

Classification performance of the interpretable hierarchical classifier is high enough not to limit interpretability. However, it is still considerable lower than current methods such as random forests. Also, our focus in this project was not on a high classification performance. Therefore we expect classification performance can be improved. For example, to discretize gene expression values we used a rather ad hoc approach, but the discretization might be improved by using the threshold between high and low expression with the highest information gain. Information gain is also used for selecting the best threshold for discretization in the ID3 algorithm for training decision trees [23].

Additionally, our use of logic formulas might be improved. The most complex logic formulas we trained also performed the best, so even more complex formulas might perform even better. At the top classifier a larger logic formula would probably impair interpretability, as the number of hallmarks it can select is only 10, and selecting more than four hallmarks could decrease their specificity. However, on the pathway level many more informative features might be available, so training larger formulas there could improve classification performance while maintaining interpretability. Also, this would increase the number of genes, which agrees with the expectations of the expert biologists.

**Improving the Visualization**

If the visualization is used in further work, it might require improvement on the points suggested by evaluation with biologists. The confusion about pathways being activated can be resolved by changing the visualization. The green lines from from pathways to hallmarks and from hallmarks to the top level classifier should be black. Red lines from pathways to hallmarks could be transformed by negating all its inputs, switching red lines to green and green lines to red, and negating its output, making it positive, so it can be colored black. In this way we present a equivalent logical structure that is more consistent in its visualization.

Another improvement to the visualization is to show the expression of more genes by using a heatmap. Heatmaps are commonly used in bioinformatics to display gene expression data, for example in [4]. A heatmap displaying all the genes in a pathway, not just the genes select by its base classifier, could give expert users insight into the activity status of that pathway.
Training the Complete Hierarchy at once

All the nodes of a level are trained independently of each other. We argued that the nodes of a level can represent different aspects of the classification problem because they can employ different inputs. But, they might not do so, so training the classifier as a whole might be advantageous. That is, we could optimize the logical formulas globally instead of locally at node. The global optimization might be implemented using integer linear programming or evolutionary computing. Both would require stating the classifier as a big logic circuit. This can then be optimized with integer linear programming using the same kind of constraints used in optimizing the logic formulas in Section 4.2. It remains to be seen whether this results in a problem that can be solved in a reasonable amount of time, or the problem size first needs to be decreased by, for example, feature selection.

Alternatively, an evolutionary computing approach [36] could also be employed to find globally near optimal logic formulas. Individuals would be hierarchical classifiers with particular logical formulas, the fitness function would be accuracy, and mutations would change the wiring between the logic. Recombination could be implemented by selecting the logic formula at a node randomly from one of the parents.

Additionally, the hierarchical structure of the classifier is reminiscent of artificial neural network, with levels corresponding to layers, and nodes with neurons. So, training procedures for artificial neural networks might be adapted to train the hierarchical classifier.
A GO-terms associated with Hallmarks

We associated high-level Gene Ontology terms [16] with the hallmarks of cancer [9, 10] as follows:

**Sustaining Proliferative Signaling**: cell proliferation (GO:0008283), cell growth (GO:0016049), cell cycle (GO:0007049), cell division (GO:0051301), positive regulation of cell proliferation (GO:0008284), positive regulation of cell growth (GO:0030307), positive regulation of cell cycle (GO:0045787), positive regulation of cell division (GO:0051781)

**Evading Growth Suppressors**: negative regulation of cell proliferation (GO:0008285), negative regulation of cell growth (GO:0030308), negative regulation of cell cycle (GO:0045786), negative regulation of cell division (GO:0051782)

**Resisting Cell Death**: programmed cell death (GO:0012501), regulation of programmed cell death (GO:0043067),

**Replicative immortality**: cellular senescence (GO:0090398), telomere organization (GO:0032200),

**Sustained Angiogenesis**: angiogenesis (GO:0001525)

**Tissue Invasion and Metastasis**: cell adhesion (GO:0007155), epithelial to mesenchymal transition (GO:0001837), locomotion (GO:0040011),

**Genome Instability**: DNA repair (GO:0006281), DNA integrity checkpoint (GO:0031570),

**Tumor-Promoting Inflammation**: inflammatory response (GO:0006954)

**Reprogramming Energy Metabolism**: metabolic process (GO:0008152)

**Evading Immune Destruction**: immune response (GO:0006955)
B Interview Questions

These are the questions we asked from participants in the evaluation with biological experts. The background question were asked at the beginning of the evaluation experiment, all other questions were asked at the end of the experiment.

Background
What is your experience with:
- gene expression data
- breast cancer
- subtypes in breast cancer
- differentially expressed genes
- pathways and pathway databases (KEGG/NCI-PID/Reactome)

Concepts
How would you describe the concept of a pathway in one sentence?
How would you describe the concept of a hallmark in one sentence?
Do you feel you understand what the is represented in the visualisation?
Does the assignment of pathways to hallmarks seem sensible to you?
Does the assignment of proteins to pathways seem sensible to you?
Do you feel you understand what the tree in the visualisation represents?
Do you feel you understand what a single box in the visualisation represents?

Usability
Do you find the visualisation simple to use?
What features of the visualisation did you find especially useful? Which not so?
Are there parts of tool that perceive as an annoyance? Which ones?
Are there features you miss? (Heatmap, etc.)
Do you have suggestions for improving visualisation?
Do you have suggestions for improving user interface?

Insights
Is there anything you already knew about the difference between the two subtypes that is confirmed by any two of the tools?
Are there any follow-up experiments suggested by what you learned?
Are there hypotheses you can formulate based on what you learned?
Bibliography


