INSTITUT FÜR INFORMATIK
Algorithmische Bioinformatik

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Identifying functional modules in *Arabidopsis thaliana* root development

Identifizieren funktionaler Module in der Wurzelentwicklung von *Arabidopsis thaliana*

Felix Grünewald

Bachelorarbeit

Beginn der Arbeit:	04. Dezember 2017
Abgabe der Arbeit:	14. März 2018
Gutachter:	Prof. Dr. Gunnar W. Klau
	Prof. Dr. Rüdiger Simon

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 14. März 2018

Felix Grünewald

Abstract

Arabidopsis thaliana has already been used as a model organism for higher plants for many years with its genome being fully sequenced around the turn of the millennium. Even for this rather simple organism, the complete biological processes are far from being fully understood. With extensive research scientists could create networks that can describe interactions between proteins, several of which we are going to incorporate into our work. When paired with data on differential expression, one is able to find functional modules, which are subgraphs in the interaction networks, that represent important pathways or processes. We used experimental data together with online resources and a set of different tools to come up with an automated workflow to extract these functional modules. Our p-values came from from a transcriptomics and a phopshoproteomics analysis based on genotypes and peptide treatments that are important for the maintenance and differentiation of stem cells in the root apical meristem of *Arabidopsis thaliana*. Once we have obtained statistical parameters from their distribution, we combine them with a protein-protein interaction network to discover subgraphs of interacting, differentially expressed genes. Afterwards we take theses results and use them to get an enrichment for our modules by comparing the genes to known data on gene ontology. By doing so, we are able to link the network's topology to actual biological functionality. Once the workflow has been executed, its outputs will be organized in such a way that they easily be visualized by using a specialized application called eXamine.

The workflow is also well suited for exploratory purposes in cases where little to no prior knowledge of the data exists due to its top-down approach. We will explain each of the steps in detail along with the theoretical concepts behind them and present exemplary use cases. Several protein-protein interaction networks were tested to find differences or similarities and potential advantages and disadvantages are discussed. We could also access an array of p-value sets to compare towards each other. Additionally we are explaining the troubles that may arise from them. To conclude this work we are going to take a brief look at a more bottom-up workflow taken from a textbook that can handle the same type of input data. When using the alternative method it is essential to know beforehand which processes might be interesting, but to have a second approach also gave us ideas for a few improvements which could be added to our workflow in the future.

Contents

1 Introduction

1.1 *Arabidopsis thaliana* as a model organism

Arabidopsis thaliana is a small, herbaceous plant in the eudicot Brassicaceae (mustard) family. By the end of the previous century, it has already been established as a commonly used model organism due to its short generation time, its small, diploid genome with around 135 million base pairs across five chromosomes, as well as its easy use for genetic transformation mediated by *Agrobacterium tumefaciens*. The first complete sequencing of *Arabidopsis thaliana* was finished by the end of 2000 [1, 2]. Today databases such as The *Arabidopsis* Information Resource (TAIR) have information about more than 35,000 genes available [3, 4].

1.2 The root development of *Arabidopsis thaliana*

Stem cells are cells in multicellular organisms that have the ability for self-renewal (regeneration) and can produce daughter cells for one or multiple kinds of specialized tissues (differentiation) through mitosis. In higher plants the tissues containing stem cells are called meristems and can be found in many different parts of the organism for the purpose of growth and self-repair. For the root development, stem cells are located in the root apical meristem (RAM) which is established during embryogenesis and remains inside the primary root after germination. The RAM is protected outwards by columella cells and the root cap. Inside the RAM there are vascular, cortex/endodermal, epidermal/lateral root cap and columella initials, that form the stem cell niche, which has the quiescent center (QC) in its middle [5], see Figure 1. Through asymmetric divisions these initials generate daughter cells for the surrounding tissues simultaneously in the distal (facing towards the tip) and proximal (facing away from the tip) direction while regenerating themselves. The cells in the QC on the other hand only possess low mitotic activity and are responsible for the maintenance of the stem cell initials through short-range signals [6].

Research has already identified several factors that are responsible for the balance between stem cell maintenance and differentiation of the initials in the RAM [7]. The regulatory pathways related to this thesis involves the transcription factor WOX5 (WUSCHEL-RELATED HOMEOBOX 5) [8]. As its name suggests, it has a similar function as WUS (WUSCHEL), which can be found not in the root, but in the shoot apical meristem. It has been observed that there is a link between WOX5, the peptide CLE40p (CLAVATA3/EMBRYO-SURROUNDING REGION 40) and the receptor kinase ACR4 (ARABIDOPSIS CRINKLY4) in the differentiation of columella stem cells (CSC) distal of the quiescent center to columella cells (CC), that are in turn distal to the CSCs [9]. One visual indicator of cell differentiation is the presence of starch granules, which could be found even in CSCs in *wox5-1* loss-of-function mutants, showing that WOX5 can maintain the CSC population. CLE40p acts antagonistically to WOX5 through ACR4. In presence of CLE40p, the expression of ACR4 is increased, which leads to a down regulation of WOX5 levels, therefore allowing for differentiation of the CSCs outside the root apical meristem.

Current models suggest that there are one or more other unknown factors involved since *wox5-1/cle40-2* double loss-of-function mutants could not be explained with previous assumptions [10]. One possible way of discovering such factors could be done through computational network analysis, for which we will provide a pipeline that identifies functional modules based on differential analysis of genome-wide expression data (transcriptomics and phosphoproteomics) using known protein-protein interaction networks of *Arabidopsis thaliana*.

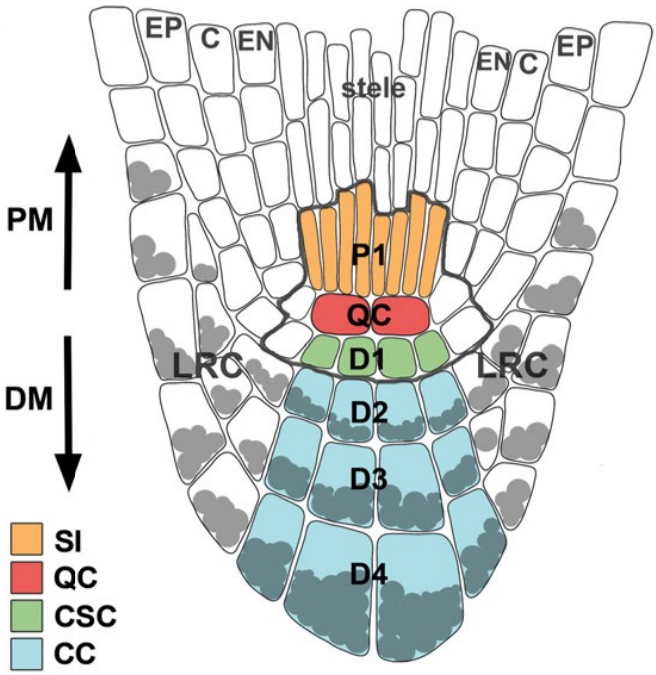

Figure 1: Schematic of the root tip of *Arabidopsis thaliana*. Growth occurs from both the distal and proximal meristems (DM/ PM). The thick outlines shows the root apical meristem with the quiescent center (QC) in the middle. D1 are the (distal) columella stem cells (CSCs), while D2 - D4 are the columella cells (CCs) with starch granules. Proximal of the QC are the stele (or vascular) initials (SI) - from [9]

1.3 Protein-protein interaction networks and functional modules

A wealth of genomic data becoming available around the turn of the millennium, such as the genome of *Arabidopsis thaliana* [2], giving rise to the new field of *functional genomics* [11]. Now that the genes have been identified (also referred to as the *post-genomic era*), it became of interest how they interacted with each other and which biological function they served. To get from the genome to a protein-protein network (interactome), several different techniques can be used [12]. The interactions can for example be directly obtained from the observation of pairwise interactions of proteins or indirectly through the coexpression of genes with either temporal, spatial or other conditional restrictions. When looking at such networks from a topological standpoint, it is easy to imagine that proteins that show a strong interconnectivity are part of or related to similar functions. These highly-connected subgraphs are commonly referred to as *(functional) modules* [13, 14]. When adding experimental data on differential expressions, we might also use the term *active modules*, because the terms are (more) active in one group than in the other.

Since many of the genes in eukaryotic organisms are orthologous, which means that they are derived from a common ancestor, gene ontology (GO) has been established with the goal to provide a vocabulary for core biological functions that are shared between species [15]. This way one is able to interfere a protein's function or interactions by comparing it to potential orthologs based on nucleic or amino acid sequences.

The three ontologies that were introduced by the GO Consortium [16] are:

- **Molecular Function** - Describes what a gene product does without specifying where or when this event takes place.
- **Biological Process** - Describes the biological objective to which a gene product contributes. This is related to, but not equivalent to, biological pathways, which usually consists of multiple processes in succession.
- **Cellular Component** - Describes the place in the cell where a gene product can be found together with others.

1.4 Objective

This work aims to explore functional modules in *Arabidopsis thaliana* by combining experimental transcriptomics and phosphoproteomics data with different protein-protein interaction networks. To achieve this we created an automated, yet customizable workflow that will generate annotated modules that can be visualized using eXamine. We will describe in detail the necessary steps and present preliminary findings based on the different data that was used. We will also discuss the current limitations of our approach and briefly compare it to an alternative workflow.

2 Materials & Methods

2.1 Sampling and processing of the experimental raw data

For this thesis two datasets were provided containing transcriptomics and phosphoproteomics data from *Arabidopsis thaliana* respectively. Samples were obtained from wild-types (strain *Columbia*) and *acr4-2* knockout mutants, both treated with CLE40p peptide and a control that only contained a phosphate buffer (labeled GM from here on) for a total of four different groups. Each group was supported by three or four repeats and will be referred to by the following notation.

Genotype:	*Columbia*	*acr4-2*
without treatment	Col-0_GM	*acr4*_GM
with treatment	Col-0_CLE40p	*acr4*_CLE40p

Table 1: Labels used for referencing the different sample groups

2.1.1 Transcriptomics data

The sampling for the data was done at the Institute of Developmental Genetics at the Heinrich-Heine-Universität in 2017. First, seeds were sown on plates with mesh and transferred for three hours on a medium that had CLE40p treatment and the control medium without peptide. Columella stem cells and columella cells were sampled from the root tips after. Then the RNA was sequenced for each sample and the amount of cells in a sample was estimated to get the average expression values per cell. In total the data of four repeats were used for both Col-0_GM and Col-0_CLE40p as well as three repeats for both *acr4*_GM and *acr4*_CLE40p.

To get the p-values, the R software environment for statistical computing and graphics [17] has been used. With the DESeq2 [18] package, which is available as part of the Bioconductor tool kit [19], the expression values were normalized. For each of the samples a single normalization factor s_j from the median-of-ratios method was used with i enumerating the different genes and j enumerating the samples in a matrix K of all expression values:

$$s_j = \underset{i:K_i^R \neq 0}{\text{median}} \frac{K_{ij}}{K_i^R} \text{ with } K_i^R = \left(\prod_{j=1}^{m} K_{ij}\right)^{1/m}$$

From the normalized data the means for each of the four sample groups were calculated. For the fold changes (FC) Empirical Bayes shrinkage [20] was used so that weak, noisy signals from very low read counts would not overshadow strong signals with high read counts. With the fold changes estimates, the p-values for the differential expression of single genes were obtained through pairwise and multifactor tests. In total the following nine sets of p-values were provided.

Set#	Label	Test type	FC Comparisons
1	t_Col-0_GM_Col-0_CLE40p	pairwise	Col-0_GM vs. Col-0_CLE40p
2	t_acr4_GM_acr4_CLE40p	pairwise	*acr4-2*_GM vs. *acr4-2*_CLE40p
3	t_Col-0_GM_acr4_GM	pairwise	Col-0_GM vs. *acr4-2*_GM
4	t_Col-0_CLE40p_acr4_CLE40p	pairwise	Col-0_CLE40p vs. *acr4-2*_CLE40p
5	t_mf_genotype	multifactor	all Col-0 vs. all *acr4-2*
6	t_mf_treatment	multifactor	all untreated vs. all CLE40p
7	t_mf2_Col-0_treatment	multifactor	Col-0 Genotype vs. GM treatment
8	t_mf2_Col-0_acr4	multifactor	(Col-0 + 7) vs. (*acr4-2* + 7)
9	t_mf2_GM_CLE40p	multifactor	(Untreated + 7) vs. (CLE40p +7)

Table 2: Labels used for referencing the different transcriptomic p-values. For 5 & 6 the samples were combined by either the genotypes or treatments. For the samples 8 & 9 the same steps as for 5 & 6 were repeated, but this time 7, Col-0 Genotype vs. GM treatment, was factored in as a correction factor as well.

2.1.2 Phosphoproteomics data

Again the sampling was done at the Institute of Developmental Genetics at the Heinrich-Heine-Universität in 2017. For this experiment the same Col-0 and *acr4-2* strains were used, but this time grown in liquid culture with three repeats for each combination of genotype and treatment. The treatment with CLE40p only lasted around 5 - 10 minutes. The phosphoproteomic analysis was then carried out by the Department of Plant System Biology at Hohenheim University employing similar methods described as described in [21]. To obtain the fold changes the R-based program cRacker [22] was used which obtained label-free ion intensities. These have been normalized using the fraction of total and then averaged before being mapped to the different proteins.

$$\text{Fraction of total } (x_i) = \frac{\text{ion intesity}}{\sum \text{all ion intesity}}$$

For the p-values a set of adjusted p-values were provided from the cRacker package. For those, the CLE40p response was measured by subtractring the log intensities of each genotypes treated minus the log intensities of each genotypes. To obtain raw p-values, we used the same pairwise comparisons made for the transcriptomics data in the previous step. Since cRacker matched short amino acid sequences to proteins, we usually had p-values for multiple parts of a protein, in which case we chose the lowest overall p-value.

Set#	Label	Test type	FC Comparisons
1	p_Col-0_GM_Col-0_CLE40p	pairwise	Col-0_GM vs. Col-0_CLE40p
2	p_acr4_GM_acr4_CLE40p	pairwise	*acr4-2*_GM vs. *acr4-2*_CLE40p
3	p_Col-0_GM_acr4_GM	pairwise	Col-0_GM vs. *acr4-2*_GM
4	p_Col-0_CLE40p_acr4_CLE40p	pairwise	Col-0_CLE40p vs. *acr4-2*_CLE40p
5	p_genotype_ttest	pairwise	treatment effects on Col-0 vs. *acr4-2*
6	p_genotype_anova	ANOVA	treatment effects on Col-0 vs. *acr4-2*

Table 3: Labels used for referencing the phosphoproteomic p-values. The p-values for 1 - 5 were obtained by using R's two-sample t-Test, while 6 used the cRacker's ANOVA approach. The log intensities for 5 & 6 were obtained by subtracting the control (GM) values from the treatment (CLE40p) averages for both genotypes.

2.2 General Workflow

With the obtained p-values we want to implement a process that will take this data along with additional online resources as input to automatically try to discover functional modules. We employ a top-down approach that starts with an uninformed topology analysis of a weighted protein-protein interaction network. In 4.2 we will also briefly discuss a more informed alternative.

For the following steps we will require the scripts that are available in this work's GitLab repository: `https://gitlab.cs.uni-duesseldorf.de/klau/BSc-thesis-network-Arabidopsis`

We will shortly discuss the three steps (see Figure 2) that are required to get from the raw p-values to the desired output before going into more detail in the upcoming subsections. When valid output has been generated for a certain input, the function calls will be skipped when the script is executed again.

1. **Beta-Uniform Mixture (BUM) model** - To get good results for our functional modules, we want to separate the good p-values from occurring noise by iterating all inputs and finding their BUM model parameters.

2. **Best-scoring subgraph** - In this step one protein-protein interaction network and the experimental p-values are used to find the best-scoring subgraph. By varying the false discovery rate (FDR) cut-offs, modules of different sizes can be found.

3. **Functional enrichment** - Lastly, we combine our module data with GO enrichment by cross-referencing the nodes to an existing GO map. Here we also generate the files needed for visualization.

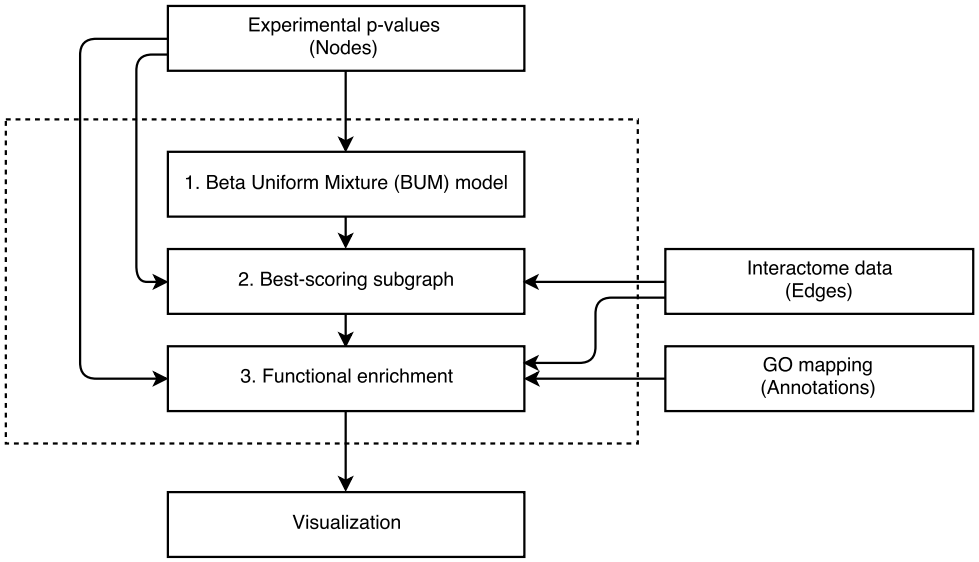

Figure 2: Flowchart of our production pipeline. The order of the steps as well as the required inputs for each step are illustrated. As the results from one step are required in the following one, it is important that they are successfully generated before moving on.

2.2.1 Necessary prerequisites

All steps were implemented and tested on a virtual machine running Ubuntu 16.04 LTS. Before the scripts can be run, the following software needs to be installed:

- R/Rscript 3.4.3 [17] with the following packages:
  - Bioconductor 3.6 [19]
  - BioNet [23]
  - GO.db [24]
  - topGO [25]
- heinz [26] - `https://github.com/ls-cwi/heinz` [1]
  After the installation `heinz` needs to be added to the path by using:
  `export PATH=~path_to_heinz/build:$PATH`
- eXamine [27] - https://github.com/ls-cwi/eXamine [2]
- Python 3.6 or higher

[1] Software required to run heinz is detailed in the repositories' readme. An academic or commercial license for IBM ILOG CPLEX is required

[2] Oracle JDK 8/ OpenJDK 8 or higher and Maven are required to build eXamine from source

2.3 Beta-Uniform Mixture (BUM) Model Fitting

To estimate the false positives and negatives in our transcriptomics and phosphoproteomics p-values, we fit a beta-uniform mixture model to our data. If the expression levels/ protein intensities were unaffected by the treatment, we would expect the p-values to follow a flat, uniform distribution (null hypothesis). However, if there are observable differences, then we would expect the p-values not to be uniformly distributed (alternative hypothesis). Even more so, we would expect that many of our p-values were close to 0 due to their heightened significance. In that case we can expect the distribution of our p-values to be composed of a mixture of a $B(a,1)$ and a uniform$(0,1) \equiv B(1,1)$ distribution [28] with a beta distribution being described as

$$B(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}$$

where $\Gamma(x)$ is the gamma function and a and b are real positive shape parameters.

2.3.1 Mathematical & algorithmic background

The beta-uniform mixture distribution $B(1,1) + B(a,1)$ can be simplified for $x \in (0,1]$ to the following probability density function:

$$f(x|a,\lambda) = \lambda + (1-\lambda)ax^{a-1}$$

λ is the mixture parameter and can be understood as the uniform distribution of noise, also referred to as the null component. a is the shape parameter of the alternative component, the $B(a,1)$ distribution. It asymptotes the y-axis and then decreases monotonically.

When approximating the values for λ and a algorithmically with BioNet [23], maximum-likelihood estimates through numerical optimization are used. This yields the values $\hat{\lambda}$ and $\hat{a}$. We can infer an upper bound for the noise π by adding $\hat{\lambda}$ and the minimum of the alternative component:

$$\pi = \hat{\lambda} + (1-\hat{\lambda})\hat{a}$$

This allows us to partition our distribution into signals above and noise below the horizontal line defined by $y = \pi$. Additionally we can assume $x = \tau$ as the vertical line representing the significance threshold. As shown in Figure 3, we get four different sectors representing the possible outcomes of a hypothesis test. Instead of selecting τ directly, we can specify an upper bound $\tilde{\alpha}$ for the false discovery rate (FDR) so that $\hat{\text{FDR}}_{ub}(\tau) \leq \tilde{\alpha}$. To describe $\hat{\tau}$ as a function of $\tilde{\alpha}$ using our estimates $\hat{\lambda}$, $\hat{a}$ and the derived $\hat{\pi}$, this formula is used:

$$\hat{\tau}(\tilde{\alpha}) = \left(\frac{\hat{\pi} - \tilde{\alpha}\hat{\lambda}}{\tilde{\alpha}(1-\hat{\lambda})} \right)^{1/(\hat{a}-1)}$$

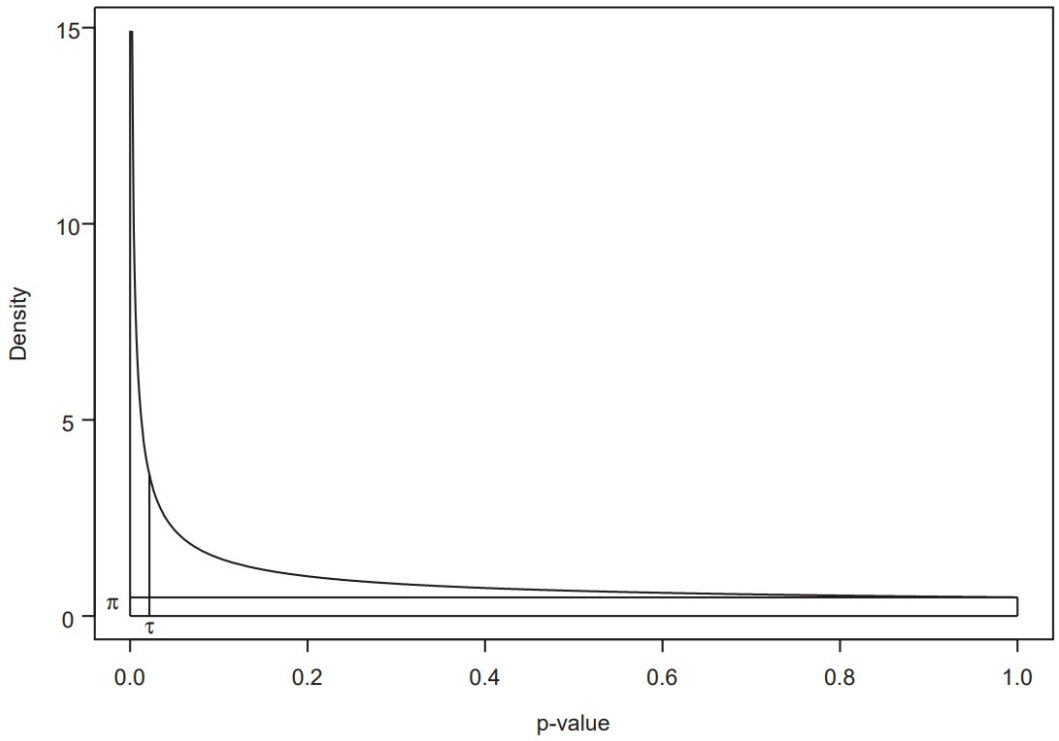

Figure 3: Schematic illustration of the BUM distribution. The intersection of noise level π and significance threshold τ divides the distribution in four sectors. Upper-left: true positives, upper-right: false negatives, lower-left: false positives and lower-right: true negatives [28].

2.3.2 Obtaining BUM model fits from experimental p-values

The first thing we want to obtain are the BUM model estimates $\hat{\lambda}$ and $\hat{a}$. To do this we first need to save our input p-values as one or more `.csv` files similarly to the example nodes in this work's repository using tab as a separator and if needed remove NaN values when no reads/ phosphopeptides were measured.

To process a single input file, we can use the function `fitBumModel(p)` of the BioNet R-package. This will create a data frame that contains the desired values along with their logarithmic likelihood and can also be used to output a histogram of the p-values overlaid with the corresponding BUM distribution.

To obtain $\hat{\lambda}$ and $\hat{a}$ from multiple files containing p-values, we can use the `bum.sh` shell script. We need to specify our output to be located in the same folder as the p-values. For the example data, this means:

```
bash bum.sh Input/Nodes/bum.txt
```

The script will now iterate over all `.csv`-files in `Input/Nodes` and try to fit a BUM

model to the p-values. The values $\hat{\lambda}$, $\hat{a}$ and the log-likelihood are stored in the output file and are required in the next step. Additionally, the histograms are saved in `Input/Nodes/Graphs`.

The script will skip over all files for which the BUM model fit has been already calculated, so if changes are made to the p-value files, they should be saved under a new name or the corresponding rows in the output need to be removed.

2.4 Calculation of the modules using Heinz

Once we have obtained the beta-uniform mixture (BUM) model parameters, we use it together with the p-values and a reference protein-protein-interaction (PPI) network in heinz ("heavy induced subgraphs") [26] to find functional modules. For this work the following PPI networks were tested:

Name	#Interactions	Citations
AINM [1]	11,374	[29]
BioGRID [2]	35,646	[30, 31]
AraNet [3]	341,821	[32]

Table 4: Labels used for referencing the protein-protein-interaction (PPI) networks. AINM uses a pipeline used for yeast two-hybrids (Y2H) [33] to find reliable binary protein–protein interactions. BioGRID has curated information from a large collection of primary sources. AraNet is a probabilistic network using data from orthologous genes in other species to estimate interactions. In this work we only used the unfiltered data from its benchmark set.

2.4.1 The scoring function in heinz

When we combine our p-values with the interactome data, we can regard each protein that has a p-value as a node in a complex network whereas the interactions represent the edges. Using our data, we want to find out which of the genes are differentially expressed, which means they are in any form amplified or reduced based on the genotype or our treatment as described in 2.1. Based on a given beta-uniform mixture (BUM) model fit, heinz [26] will partition the data into signal and noise with a certain threshold that is defined as the false discovery rate (FDR). To do this, heinz scores the p-values with an adjusted log likelihood ratio given as

$$S^{FDR}(x) = \log\left(\frac{ax^{a-1}}{a\tau^{a-1}}\right) = (a-1)\left(\log x - \log \tau(\tilde{\alpha})\right).$$

$\tau(\tilde{\alpha})$ here is the threshold p-value based on the chosen FDR value and the BUM model fit parameters λ and a. There is a change of sign once the observed p-value x becomes

[1] http://interactome.dfci.harvard.edu/A_thaliana/index.php
[2] https://thebiogrid.org/ (v3.4.157)
[3] http://www.functionalnet.org/aranet/

bigger (negative) or smaller (positive) than the threshold value. If a = 1 all p-values are subject to a uniform ($B(1,1)$) distribution and therefore the scores are always 0.

To see this in practice, we anticipate some of the results for an example for the resulting threshold from the expression data for untreated genotypes (t_Col-0_GM_acr4_GM):

$$\hat{\lambda} \approx 0.543, \hat{a} \approx 0.174, \text{FDR} = 0.08$$
$$\hat{\pi} = \hat{\lambda} + (1 - \hat{\lambda})\hat{a} \approx 0.543 + 0.457 \cdot 0.174 \approx 0.623$$
$$\hat{\tau}(\tilde{\alpha}) = \left(\frac{\hat{\pi} - \tilde{\alpha}\hat{\lambda}}{\tilde{\alpha}(1 - \hat{\lambda})} \right)^{1/(\hat{a}-1)} \approx \left(\frac{0.623 - 0.08 \cdot 0.543}{0.08 \cdot 0.457} \right)^{-2.193} \approx 0.072$$

In this example all p-values smaller than the threshold of 0.072 would score positively, while all of the bigger p-values would receive a negative score. While the scoring of the nodes is independent from the PPI network, the resulting modules are not.

2.4.2 The Prize-Collecting Steiner Tree (PCST) Problem

Given the edges in the PPI network and our now scored nodes, heinz starts identifying the best-scoring (also called maximum-weight) connected subgraph. The total score for a given subgraph is obtained by adding all the individual scores for each node together. Since the insignificant (which are the ones higher than our threshold τ) p-values result in a negative score, we want to avoid penalties as much as possible. However in some cases it is beneficial to incorporate a negative node in the subgraph if it allows us to reach an otherwise unreachable positive node with a net gain. To solve this NP-complete problem more efficiently, it gets transformed to another problem: the Prize-Collection Steiner Tree (PCST) problem to eliminate negative weights. To get from finding maximum-weight connected subgraph to the best PCST, the node with the most negative score is sought. Afterwards each node's score gets incremented by the positive equivalent of this score to make up the p̈rizes.̈ Additionally the negative value is added as costs to every edge.

This reformulated problem's objective is now to maximize the gain by visiting highly prized nodes while using as little edges as possible. This is solved by heinz by expanding upon the algorithm used in a program called dhea (district heating) [34]. One of the expansions is the ability to obtain k similar, yet suboptimal solutions, because the resulting modules might give some additional insight as well as add otherwise undiscovered nodes to the tree.

2.4.3 Automated process for varying the false discovery rate (FDR)

For this step three different inputs are required: First, we need the BUM model fits calculated in 2.3.2. Because the output of this step is located in the same folder as our p-value files, we can iterate all files to which we could successfully fit a BUM model while skipping those with a log-likelihood of 0.0, because then we cannot distinguish the signals from noise. The second input is the chosen PPI network, that is a tab-separated file that only contains the two interaction partners without additional information. The last input

is a file that contains the desired FDR values in exponential or decimal notation. For this work we generally kept the FDR between 1E-5 and 5E-12, but when the networks were still too small or too big, we made adjustments to cover values between 1E-1 and 1E-15. For the example data and the BioGRID PPI network, we would use the following call:

```
bash enrichment.sh Input/Nodes/bum.txt Input/BioGRID.txt Input/fdr.txt
```

The output will be saved in a folder called `Heinz` which contains subfolders for the different p-value sets in which the scores and an image of the found network are stored in correspondence to the FDR-values. Combinations of FDR and p-value file names that have already been calculated will be skipped in subsequent calls.

2.5 Functional enrichment with topGO

When we have successfully identified one or more modules with heinz, we can have a look at whether we can find biological context in them. With our current knowledge we know which protein-protein-interaction are supported by our data of differentially expressed genes, but it is also desirable to find out which purpose the interaction serves.

2.5.1 Gene Ontology (GO) modules

In order to obtain the Gene Ontology (GO) enrichment for the discovered modules, the R package topGO [25] is used. To prepare, an existing GO map is compressed to a single line per gene along with the experimental p-value. All GO terms related to a gene are split across the three different GO domains: Molecular Function, Biological Process and Cellular Component. We also add the information whether or not the gene is part of the modules obtained via heinz, because package allows us to input a list of interesting genes to be featured in the output for several statistical tests. In this work, the so called classic test (each GO domain is tested individually) with Fisher statistics has been used. This will generate a list of up to 20 of the GO terms with the highest significance per domain that are then merged into one file for further analysis.

2.5.2 Putting it all together

For the enrichment we need the output generated by heinz as well as a mapping of the proteins to GO terms. In this work we have used ATH_GO from he Arabidopsis Information Resource (TAIR) [35] and written `merge.py` with specifically this map's structure in mind. If other maps were to be used, then the column indices for the following information would need to be specified in the file: locus name, GO ID and the GO domain (C, F or P). A script call needs to reference the GO map file and the interactome that should be used:

```
bash enrichment.sh Input/ATH_GO_GOSLIM.txt Input/BioGRID.txt
```

This will first find all unique GO terms in the map file and their ancestors by comparing with the dataset of GO.db [24] for R. Afterwards the script will iterate all the heinz outputs that were generated and will find the best GO terms for each domain. In a folder labeled `Output` the data for visualization will be stored separated by dataset and false discovery rates (FDR). The other output folders `Enrichment` (for topGO results), `eXamine` (for `.exm` files for visualization with older versions of eXamine, see 2.6) and `GO` (for the GO terms of the nodes) are currently used to store intermediate results. As with the other scripts, this will also skip over successfully enriched modules.

2.6 Visualization with eXamine

Finally, the results can be visualized with eXamine [27] using the latest release of a standalone version[3] for further analysis. With it one can select interesting GO modules from a list of annotated terms and the displayed graph will reorganize itself in such a manner that all biologically connected nodes are outlined by a contour. This can be done for one or multiple GO terms at a time. Also a single gene can be selected and all the interactions of it are highlighted. Support for multiple datasets is enabled and so it is possible to compare results for different false discovery rate (FDR) values as well as suboptimal solutions. In the last step of the script described in 2.5.2, the files that can be used with eXamine have been created. The required files and their contents are:

- `proteins.nodes` - A list of all proteins in the module with their respective score (see 2.4.1) and possibly additional information such as labels or URLs
- `modules.annotations` - This contains all labeled modules. However for this work the file is hard-coded since we only output a single module.
- `modules.links` - This file links each proteins to a module.
- `interactions.links` - A list of all the edges in the module(s).
- `go_and_kegg.annotations` - A selection of the best-scoring GO terms that are contained in the module(s) with annotations.
- `go_and_kegg.links` - This file links each protein to a GO term if possible.

For this subtask the workflow is very similar to the overall approach described in 2.2. We have again a subdivision into three scripts `nodes.py`, `interactions.py` and `go_modules.py` in which the results from one step are passed as input to the next one. First the nodes are gathered and linked to a single module, then the edges are selected from the interactome that was used. In the last step the GO annotations are written and a link between nodes and GO terms is established.

[3] https://github.com/AlBi-HHU/eXamine-stand-alone

3 Results

3.1 Beta-Uniform-Mixture (BUM) model fits of the data

By using the BioNet package [23] in R, we obtained a first insight into the distribution of p-values in our datasets. Differential expression was best observed between the two species in an untreated (t_Col-0_GM_acr4_GM, log likelihood: 28415.998) and treated state (t_Col-0_CLE40p_acr4_CLE40p, log l.: 44427.138), and also in the pooled results between genotypes (t_mf_genotype, log l.: 79440.775) as well as the pooled results considering the effects of the treatment on the wild type (t_mf2_Col-0_acr4, log l.: 44427.259).

Set#	Label	Nodes	Mixture $\hat{\lambda}$	Shape $\hat{a}$	log likelihood
1	t_Col-0_GM_Col-0_CLE40p	24966	1.000	1.000	0.000
2	t_acr4_GM_acr4_CLE40p	24966	0.909	0.308	846.368
3	t_Col-0_GM_acr4_GM	24966	0.543	0.174	28415.998
4	t_Col-0_CLE40p_acr4_CLE40p	24966	0.510	0.139	44427.138
5	t_mf_genotype	24936	0.431	0.107	79440.775
6	t_mf_treatment	24938	1.000	1.000	0.000
7	t_mf2_Col-0_treatment	24966	0.953	0.301	330.630
8	t_mf2_Col-0_acr4	24966	0.510	0.139	44427.259
9	t_mf2_GM_CLE40p	24966	0.909	0.308	846.429
10	p_Col-0_GM_Col-0_CLE40p	1592	1E-05	0.821	33.143
11	p_acr4_GM_acr4_CLE40p	1592	1E-05	0.895	10.227
12	p_Col-0_GM_acr4_GM	1592	1E-05	0.870	16.227
13	p_Col-0_CLE40p_acr4_CLE40p	1592	1E-05	0.807	39.186
14	p_genotype_ttest	1592	1E-05	0.841	25.348
15	p_genotype_anova	1592	0.604	0.683	24.952

Table 5: Beta-Uniform Mixture (BUM) model fits parameters for the samples. Results were rounded to three decimal places.

We observed very different distributions of p-values between most of the samples, but there were also some with very similar parameters. While not visible in the rounded results, there were small differences (<1E-5) in the values for $\hat{\lambda}$ and $\hat{a}$ for the samples t_acr4_GM_acr4_CLE40p and t_mf2_GM_CLE40p as well as t_Col-0_CLE40p_acr4_CLE40p and t_mf2_Col-0_acr4.

In the following steps all model fits with a log likelihood of 0 (t_mf_treatment, t_Col-0_GM_Col-0_CLE40p) were skipped as they did not follow the desired beta-uniform model (BUM) distribution. Here $\hat{\lambda}$ will be 1 so that the signal term in the formula $\hat{\pi} = \hat{\lambda} + (1 - \hat{\lambda})\hat{a}$ disappears completely and we only get a horizontal line instead of an asymptotic curve. The amount of noise can also be seen in the histogram outputs (Figure 4). As described in 2.4.3 a lower log likelihood requires a more generous false discovery rate (FDR) for the following step in our pipeline. For the phosphoproteomic data the log likelihood was barely above 0 and overall not usable for further analysis.

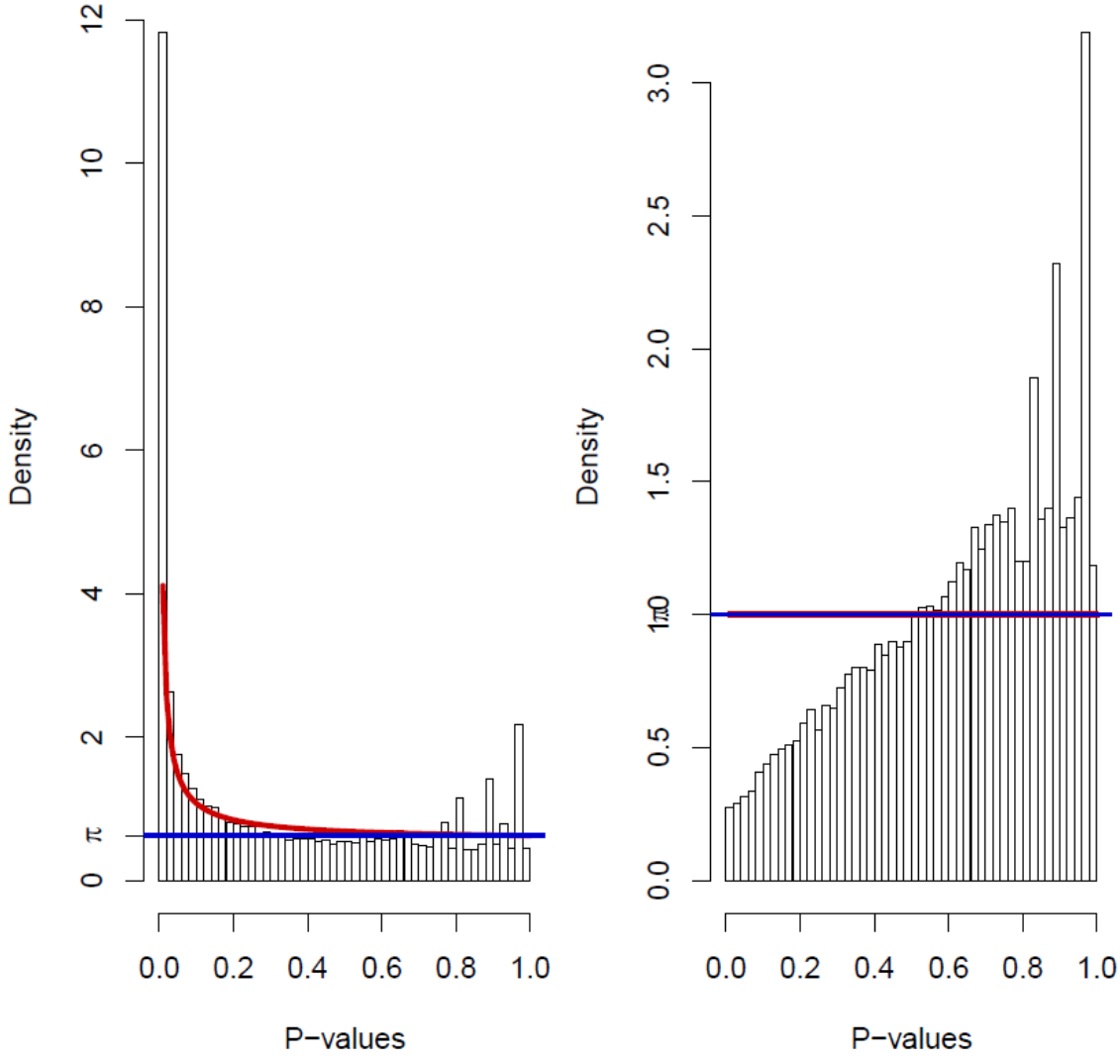

Figure 4: Two sample distributions of p-values when comparing the untreated wild-type (Col-0 GM) to the untreated mutant (*acr4_2 GM*, left) and treated wildtype (Col-0 CLE40p, right). The red line is the beta-distributed component and the blue line is the uniform component. If our sample p-values do not follow a beta-uniform mixture distribution ($\hat{\lambda} = \hat{a} = 1$), both lines converge.

3.2 Descriptive analysis of the discovered modules

After we ran heinz, we compared the varying sizes of the networks dependent on the dataset (and FDR values) used for the nodes as well as the different interactomes and briefly summarized the intermediate results. The metrics that we will use are the total number of nodes as well as the scores per module.

As discovered in the previous step, the p-values of t_acr4_GM_acr4_CLE40p and t_mf2_GM_CLE40p as well as t_Col-0_CLE40p_acr4_CLE40p and t_mf2_Col-0_acr4 followed had almost identical distribution. By using heinz we could also see that save for negligible small differences in the scores the resulting trees were identical for all FDR values and interactomes. This means that two more (t_mf2_GM_CLE40p and t_mf2_Col-0_acr4) of the remaining seven datasets could potentially be left out in further analysis.

3.2.1 Summary of the intermediate outputs from the "omic" data

The module sizes across the different datasets changed in accordance to the BUM model fits, so the datasets with the smallest log likelihood also had the lowest scores as well as the least amount of nodes for a set FDR value and interactome with the resulting trees often consisting of a single node with the highest p-value. On average the number of nodes went up in a quadratic manner compared to the score as we also added more and more hub nodes to the graph.

Because the score of each node increases linearly with the logarithm of the FDR, we observed many cases where the tree completely changed its topology once a certain threshold value has been reached. The isolated nodes that resulted from a protein with no reported interactions or only interactions with not well supported proteins were then replaced by a completely different tree with multiple interactions.

3.2.2 Summary of the intermediate outputs from the interactome data

Unlike the log likelihood for our BUM model fits, when it came to the interactomes it did not hold true that more edges meant that the the resulting networks would be bigger. In fact, for our data the medium sized BioGRID network had the best performance in every test that was run. To find out the similarity between the different networks, we have calculated the number of identical (undirected) edges.

	BioGRID (35646 Edges)	AraNet (341821 Edges)
AINM (11374 Edges)	8886 = 78%	670 = 6%
BioGRID (35646 Edges)		1417 = 4%

Table 6: Number of intersecting edges between the interactomes. The percentages relate to the smaller interactome. The intersection of all three interactomes consisted of 561 edges which corresponds to 5% of AINM.

While AINM and BioGRID were quite similar in regards to the edges that they contained, the biggest differences were found between each of them and the probabilistic AraNet

network. This also had a significant influence on the modules that were obtained as the genes included in the networks showed little overlap as well. While no benchmarks on the runtime were run, it became apparent that only the usage of AINM and BioGRID resulted in short enough runtimes for heinz to be used in a setup with many different datasets and FDR values. As finding the modules is a NP-hard problem, the runtimes for AraNet were problematic when we tried to run 30 or more tests in succession with runtimes often thirty times higher than when using BioGRID.

3.3 Analysis of the discovered GO terms

For the next step we have taken a look a closer look at the GO terms that were identified. The total number of modules that were calculated for a single interactome were between 50 and 59. As the modules in heinz were obtained by looking at differentially expressed genes, a GO function or process in the context of our experimental data should mean that the differences between the two groups led to a significant up- or downregulation of said GO domain due to differences in treatments or the genotype. Because topGO also looked at nodes that were outside of our modules, GO terms were oftentimes also found for nodes that weren't part of the modules. These terms had to be excluded in the final step to produce the output files for visualization. Using the default setting of the twenty top scoring GO terms for each domain, only two to three GO processes were left in the final output. However it is generally possible to add more enrichment to the data by increasing the desired amount of GO terms for the output and repeat the final step of our workflow.

As mentioned in 3.2.1 the false discovery rate (FDR) had a direct influence on the size of the modules. When we look at the changes of the p-values of the GO terms, no such correlation can be found. When the network gets bigger, the p-values for certain terms became smaller as more nodes were added as support while for others more unrelated nodes diluting the GO terms meant a lower significance. In this list are the most frequent terms for GO processes for each of the interactomes used in descending order by the number of their occurrence:

- AINM: mitochondrial electron transport, succinate to ubiquinone (GO:0006121), signal transduction (GO:0007165), regulation of seed germination (GO:0010029)
- AraNet: mitochondrial electron transport, succinate to ubiquinone (GO:0006121), defense response (GO:0006952), multicellular organismal development (GO:0007275)
- BioGRID: mitochondrial electron transport, succinate to ubiquinone (GO:0006121), signal transduction (GO:0007165), response to hormone (GO:0009725)

As before the results were showing that AINM and BioGRID had more similar results than AraNet, but the top scoring GO process for all interactomes was *mitochondrial electron transport, succinate to ubiquinone* (GO:0006121) which seems to be influenced especially by the CLE40p treatment of *acr4-2* mutants.

We were also interested in the variability of both GO terms as well as module nodes in certain interactomes, because we saw some of the GO terms only appear once for all of the datasets. The Venn diagrams used were generated using the web application InteractiVenn [36].

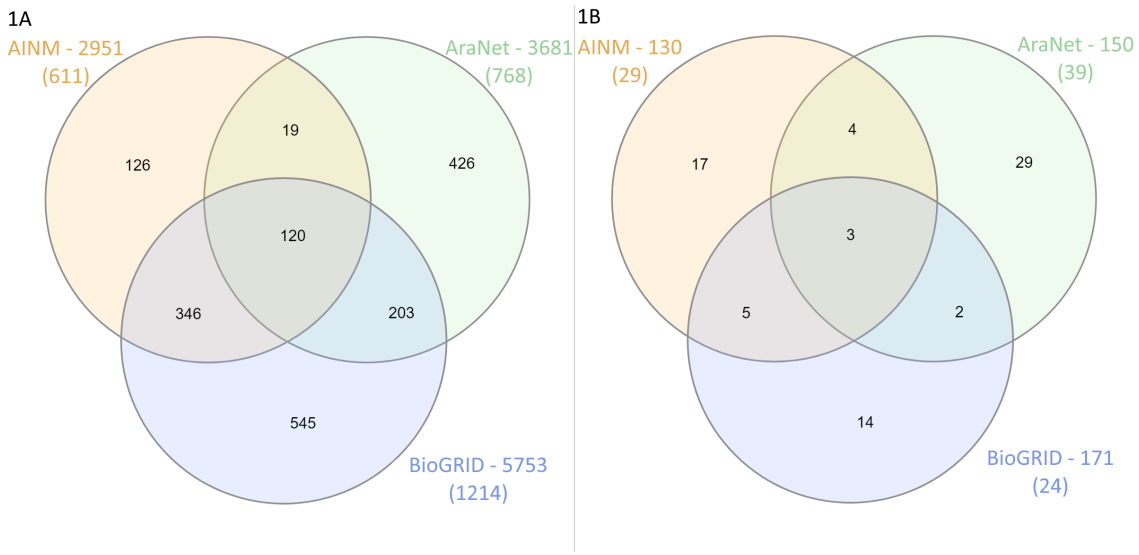

Figure 5: Intersections of nodes and GO terms. The total numbers of modules that were pooled are 56 for AINM, 50 for AraNet and 59 for BioGRID. Diagram 1A shows the intersections of unique nodes. For each interactome the total number of pooled nodes is given beside the interactome names and the number of unique nodes is in brackets. The average numbers of nodes per module therefore are 53 for AINM, 74 for AraNet and 98 for BioGRID. Diagram 1B represents the intersections of the pooled GO processes. Here the total number of pooled GO processes is given beside the interactome names and the number of unique processes is in brackets. Here the averages per module are 2 GO processes for AINM and 3 for AraNet and BioGRID topGO's default settings.

For all of the interactomes the ratio of unique to overall nodes was around 1 to 5. This might be in part explained by the fact that the modules were based on multiple datasets, so a certain variety should be expected. It can be seen that the intersections of the unique nodes are showing a similar behavior to the overlap of edges between the interactomes, so when using PPI networks with a higher similarity, the resulting modules also contained more of the same nodes. The overlap in terms of unique nodes between AINM and BioGRID is so big that only 24% of AINM's nodes were not shared between the two.

The ratio of unique GO processes to overall GO processes was very different for each interactome while the percentages of the overlap were again similar to our previous comparisons. Interestingly, while having the most nodes and GO processes, the BioGRID modules were also the most stable ones in terms of the GO processes they contained. This can be interpreted as BioGRID being the least affected by variability of the biological processes found in the modules. While the size of the networks changed greatly depending on the FDR values, the most probable GO terms seemed to be fairly conserved here.

3.4 Visualization of the obtained modules

While it is impossible to discuss all of the modules found in this work, we want to focus on giving one detailed example of a visualized module. Given all that we have learned thus far about the different interactomes used, we think that the most suitable modules that were found came from using the BioGRID interactome with the modules having the highest score in heinz and the most conserved GO processes. When it came to interpreting the modules, size was always an important factor to consider as some of the very complex, which is why we decided to crop the module in Figure 6 for illustrational purposes. The full version is included in the appendix.

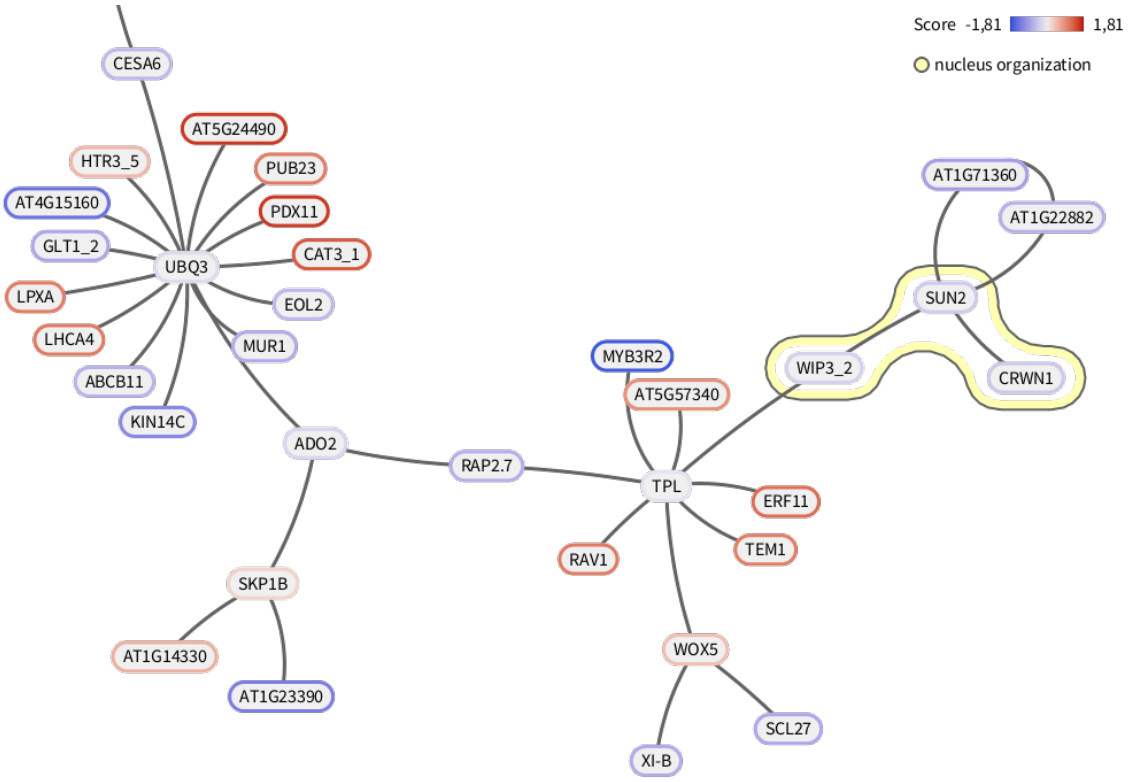

Figure 6: Visualization of a functional module in eXamine (cropped). This module has been obtained by using the t_acr4_GM_acr4_CLE40p p-values and the BioGRID interactome at an FDR of 0.08. The nodes are colored by the log fold changes after the CLE40p treatment, red: positive, blue: negative. One group of the GO process *nucleus organization* has been highlighted to demonstrate eXamine's capabilities.

In this module we can see how the visualization of the modules generally looked like. For the color gradient, we chose to use the observed log fold changes, but the nodes can also be weighted with the scores received from heinz, compare 2.4.1. In the networks generated by heinz nodes in the periphery of the module always possess a positive score, so that the p-values are lower than our margin defined by the FDR. Only in modules with a high FDR setting, we were able to find WOX5, which has been described in 1.2 and which is here connected to TOPLESS (TPL), SCARECROW-LIKE 27 (SCL27) and

MYOSIN XI B (XI B). However it has to be noted that the fold change of WOX5 expression levels is not well supported which is why this node received the most negative score. We already saw that CLE40p was not affecting WOX5 levels much in *acr4-2* knockouts [37], so it is debatable if any differential expression occurred.

4 Discussion & Outlook

4.1 Evaluation of the results

For this work we successfully created an automated workflow that employs several different techniques to output functional modules based on a combination of experimental and literature data. The workflow is also able to be resumed in case of singular faulty inputs and is open for extension in terms of working processes or parameter changes for the scripts. The outputs were tested and up to a certain size could be visualized with eXamine.

However, for the specific data at hand, we unfortunately weren't able to find many reliable or interesting modules. Oftentimes nodes relating to the same GO process showed no real interconnectivity, probably due to important nodes along the process missing from the module due to bad scores. One source of the problems was probably the use of unfiltered interactome data, especially in the case of AraNet. With more insight on the data and by limiting the number of interactions, it could be possible to increase the reliability of the results.

Another factor can also arise from the experimental data, more specifically the number of samples, is that the raw p-values often were very low, sometimes down to to E-200. This meant that very low false discover rate (FDR) values had to be selected and even with 1E-15 the modules remained too large for the BioGRID interactome. This is due to the way heinz will score each of the nodes according to the FDR value creating a big bias towards those samples that came up with the best p-values.

4.2 Comparison to an alternate workflow

As mentioned in 2.3.2 our workflow relies a lot on a suitable distribution of the raw p-values. In some cases this may even lead to finding no modules at all and sometimes no raw p-values might be available. An alternative workflow is the one described in "Auxins and Cytokinins in Plant Biology" [38]. While the tools used weren't that much different from ours, the approach is a lot more bottom-up in a sense that a process or other GO term of interest has to be specified beforehand, see Figure 7.

We have tried this approach using strict p-value cutoffs and the BioGRID interactome in an attempt to double-check some of our modules or find results where we couldn't before. We used one of the phosphoproteomics datasets (p_genotype_anova: p-value: 0.05), but weren't able to find any good modules again. We also tried two transcriptomics datasets (t_Col-0_GM_acr4_GM, p-value: 1E-8 & t_mf_genotype, p-value: 1E-15) that performed well with our workflow, but then ran into difficulties when settling for smaller

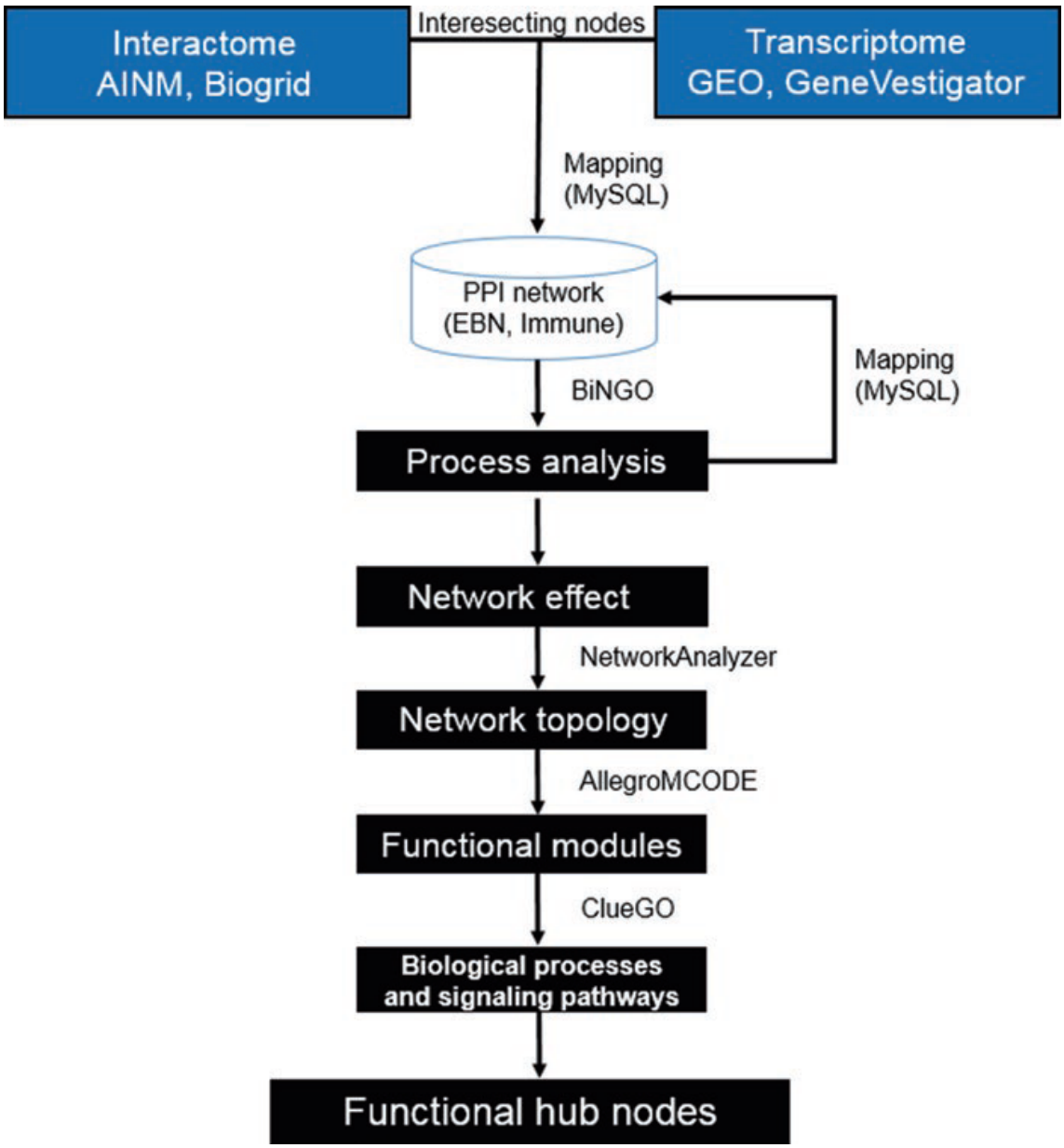

Figure 7: Alternative workflow to find functional modules as described in [38]. Here the process analysis is made before looking at the network topology, while in our workflow these steps were reversed, see Figure 2 for comparison.

GO terms to investigate, because the networks still had around 700 nodes left at that point and the GO processes did not cluster well. However these results are more likely related to the experimental data than the workflow itself, so an extensive evaluation using more datasets might be interesting.

4.3 Outlook

Probably the biggest addition that can be made to the process is the implementation of an even usability. As of now the execution of all the steps requires to follow a certain protocol as well as the installation of various required software. With build tools such as Snakemake [39] the steps can be further unified as well as simplified so that the builds always run using the same dependencies. Additionally the automation might also be able to keep the footprints of generated data small if desired or give detail files for all the steps along the way as it does now. Another addition could be to extend the workflow with DESeq2 [18] or similar tools to automate the analyze starting with the raw expression/ ion intensity data.

As it stands the workflow still relies in some parts on manual analysis, but when combined with a list of certain goals set a priori, such as interesting nodes or GO terms, we think that the results may become more reliable. In that case we could look for modules with the highest score to size ratio or which are in a certain margin for the number of nodes. Other parameters that might be interesting to change could also be the number of GO terms that were output by topGO as we have seen that sometimes only two or three processes were contained in our modules. or certain nodes. On top of that it might also be good to make searches a little easier a posteriori for a more exploratory analysis like we did in this work.

When it comes to visualization, in the current state the nodes contain only the identifier, gene name and the score. Additional information might be useful such as the inclusion of links to online resources such as Uniprot [40] or for *Arabidopsis thaliana* specifically TAIR [3]. It might also be interesting to have the outputs not only cover either the scores, the p-values or the fold changes, but also information on all three at the same time.

5 Acknowledgments

I would like to express my gratitude to my supervisor Prof. Dr. Gunnar Klau for giving me the opportunity for an interdisciplinary work. Without his guidance and persistent help this thesis would not have been possible.

I would also like to thank Prof. Dr. Rüdiger Simon from the Institute of Developmental Genetics for being this work's second assessor. Additionally, I would like to thank him and Dr. Barbara Berckmans for allowing me to work with their research data and for providing me with explanations and literature along the way.

References

[1] D. W. Meinke, J. M. Cherry, C. Dean, S. D. Rounsley, and M. Koornneef, "Arabidopsis thaliana: A Model Plant for Genome Analysis," *Science (New York, N.Y.)*, vol. 282, no. 5389, pp. 662–682, 1998.

[2] The Arabidopsis Genome Initiative, "Analysis of the genome sequence of the flowering plant Arabidopsis thaliana," *Nature*, vol. 408, no. 6814, p. 796, 2000.

[3] S. Y. Rhee, W. Beavis, T. Z. Berardini, G. Chen, D. Dixon, A. Doyle, M. Garcia-Hernandez, E. Huala, G. Lander, M. Montoya, N. Miller, L. A. Mueller, S. Mundodi, L. Reiser, J. Tacklind, D. C. Weems, Y. Wu, I. Xu, D. Yoo, J. Yoon, and P. Zhang, "The Arabidopsis Information Resource (TAIR): A model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community," *Nucleic Acids Research*, vol. 31, no. 1, pp. 224–228, 2003.

[4] T. Z. Berardini, L. Reiser, D. Li, Y. Mezheritsky, R. Muller, E. Strait, and E. Huala, "The Arabidopsis Information Resource: Making and mining the "gold standard" annotated reference plant genome," *genesis*, vol. 53, no. 8, pp. 474–485, 2015.

[5] R. Sozzani and A. Iyer-Pascuzzi, "Postembryonic control of root meristem growth and development," *Current opinion in plant biology*, vol. 17, pp. 7–12, 2014.

[6] C. Delay, N. Imin, and M. Djordjevic, "Regulation of Arabidopsis root development by small signaling peptides," *Frontiers in Plant Science*, vol. 4, p. 352, 2013.

[7] A. P. Fisher and R. Sozzani, "Uncovering the networks involved in stem cell maintenance and asymmetric cell division in the Arabidopsis root," *Current opinion in plant biology*, vol. 29, pp. 38–43, 2016.

[8] A. K. Sarkar, M. Luijten, S. Miyashima, M. Lenhard, T. Hashimoto, K. Nakajima, B. Scheres, R. Heidstra, and T. Laux, "Conserved factors regulate signalling in Arabidopsis thaliana shoot and root stem cell organizers," *Nature*, vol. 446, no. 7137, pp. 811–814, 2007.

[9] Y. Stahl and R. Simon, "Is the Arabidopsis root niche protected by sequestration of the CLE40 signal by its putative receptor ACR4?," *Plant Signaling & Behavior*, vol. 4, no. 7, pp. 634–635, 2009.

[10] S. Richards, R. H. Wink, and R. Simon, "Mathematical modelling of WOX5- and CLE40-mediated columella stem cell homeostasis in Arabidopsis," *Journal of experimental botany*, vol. 66, no. 17, pp. 5375–5384, 2015.

[11] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, "Protein function in the post-genomic era," *Nature*, vol. 405, no. 6788, pp. 823–826, 2000.

[12] B. S. Srinivasan, N. H. Shah, J. A. Flannick, E. Abeliuk, A. F. Novak, and S. Batzoglou, "Current progress in network research: Toward reference networks for key model organisms," *Briefings in bioinformatics*, vol. 8, no. 5, pp. 318–332, 2007.

[13] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761 Suppl, pp. C47–52, 1999.

[14] A.-L. Barabási and Z. N. Oltvai, "Network biology: Understanding the cell's functional organization," *Nature reviews. Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[15] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[16] Gene Ontology Consortium, "Creating the gene ontology resource: Design and implementation," *Genome research*, vol. 11, no. 8, pp. 1425–1433, 2001.

[17] R Core Team, "R: A Language and Environment for Statistical Computing," 2017.

[18] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome biology*, vol. 15, no. 12, p. 550, 2014.

[19] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang, "Bioconductor: Open software development for computational biology and bioinformatics," *Genome biology*, vol. 5, no. 10, p. R80, 2004.

[20] G. K. Smyth, "Linear models and empirical bayes methods for assessing differential expression in microarray experiments," *Statistical applications in genetics and molecular biology*, vol. 3, pp. 1–25, 2004.

[21] X. N. Wu and W. X. Schulze, "Phosphopeptide profiling of receptor kinase mutants," *Methods in molecular biology (Clifton, N.J.)*, vol. 1306, pp. 71–79, 2015.

[22] H. Zauber and W. X. Schulze, "Proteomics wants cRacker: Automated standardized data analysis of LC-MS derived proteomic data," *Journal of proteome research*, vol. 11, no. 11, pp. 5548–5555, 2012.

[23] D. Beisser, G. W. Klau, T. Dandekar, T. Müller, and M. T. Dittrich, "BioNet: An R-Package for the functional analysis of biological networks," *Bioinformatics (Oxford, England)*, vol. 26, no. 8, pp. 1129–1130, 2010.

[24] M. Carlson, "GO.db: A set of annotation maps describing the entire Gene Ontology," *R package version*, 2013.

[25] A. Alexa and J. Rahnenfuhrer, "topGO: Enrichment analysis for gene ontology," *R package version*, 2010.

[26] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller, "Identifying functional modules in protein-protein interaction networks: An integrated exact approach," *Bioinformatics (Oxford, England)*, vol. 24, no. 13, pp. i223–i231, 2008.

[27] K. Dinkla, M. El-Kebir, C.-I. Bucur, M. Siderius, M. J. Smit, M. A. Westenberg, and G. W. Klau, "eXamine: Exploring annotated modules in networks," *BMC Bioinformatics*, vol. 15, p. 201, 2014.

[28] S. Pounds and S. W. Morris, "Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values," *Bioinformatics (Oxford, England)*, vol. 19, no. 10, pp. 1236–1242, 2003.

[29] Arabidopsis Interactome Mapping Consortium, "Evidence for network evolution in an Arabidopsis interactome map," *Science (New York, N.Y.)*, vol. 333, no. 6042, pp. 601–607, 2011.

[30] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: A general repository for interaction datasets," *Nucleic acids research*, vol. 34, no. Database issue, pp. D535–D539, 2006.

[31] A. Chatr-Aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, N. K. Kolas, L. O'Donnell, S. Oster, C. Theesfeld, A. Sellam, C. Stark, B.-J. Breitkreutz, K. Dolinski, and M. Tyers, "The BioGRID interaction database: 2017 update," *Nucleic acids research*, vol. 45, no. D1, pp. D369–D379, 2017.

[32] I. Lee, B. Ambaru, P. Thakkar, E. M. Marcotte, and S. Y. Rhee, "Rational association of genes with traits using a genome-scale gene network for Arabidopsis thaliana," *Nature biotechnology*, vol. 28, no. 2, pp. 149–156, 2010.

[33] M. Dreze, D. Monachello, C. Lurin, M. E. Cusick, D. E. Hill, M. Vidal, and P. Braun, "High-quality binary interactome mapping," *Methods in enzymology*, no. 470, pp. 281–315, 2010.

[34] I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti, "An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem," *Mathematical Programming*, vol. 105, no. 2-3, pp. 427–449, 2006.

[35] T. Z. Berardini, S. Mundodi, L. Reiser, E. Huala, M. Garcia-Hernandez, P. Zhang, L. A. Mueller, J. Yoon, A. Doyle, G. Lander, N. Moseyko, D. Yoo, I. Xu, B. Zoeckler, M. Montoya, N. Miller, D. Weems, and S. Y. Rhee, "Functional annotation of the Arabidopsis genome using controlled vocabularies," *Plant physiology*, vol. 135, no. 2, pp. 745–755, 2004.

[36] H. Heberle, G. V. Meirelles, F. R. da Silva, G. P. Telles, and R. Minghim, "InteractiVenn: A web-based tool for the analysis of sets through Venn diagrams," *BMC Bioinformatics*, vol. 16, p. 169, 2015.

[37] Y. Stahl, R. H. Wink, G. C. Ingram, and R. Simon, "A Signaling Module Controlling the Stem Cell Niche in Arabidopsis Root Meristems," *Current Biology*, vol. 19, no. 11, pp. 909–914, 2009.

[38] T. Dandekar and M. Naseem, "Auxins and Cytokinins in Plant Biology," vol. 1569, pp. 165–173, 2017.

[39] J. Köster and S. Rahmann, "Snakemake–a scalable bioinformatics workflow engine," *Bioinformatics (Oxford, England)*, vol. 28, no. 19, pp. 2520–2522, 2012.

[40] The UniProt Consortium, "UniProt: The universal protein knowledgebase," *Nucleic acids research*, 2018.

List of Figures

List of Tables

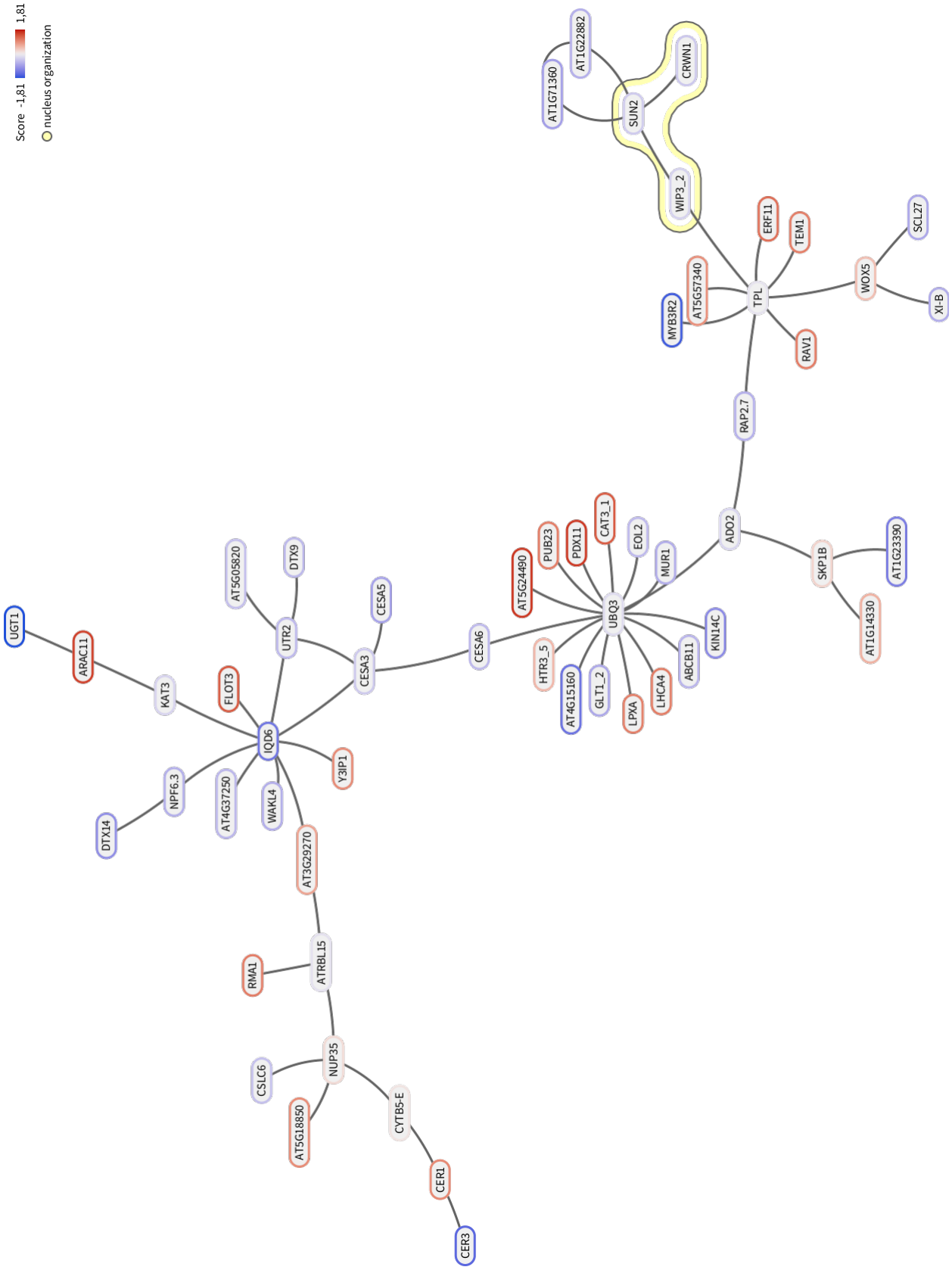

Figure 8: Visualization of a functional module in eXamine (complete)