PaIntDB: Visualizing Protein-Protein and Protein-Metabolite Interaction Networks in *Pseudomonas aeruginosa*

Javier J. Castillo-Arnemann (javier@hancocklab.com)

1 Abstract

PaIntDB is an interactive network visualization tool designed to help biologists interpret the results from high-throughput experiments. It takes a list of genes as input and outputs a network that can be explored interactively within the application. To tackle the common hairball issue that is prevalent in large network visualizations, it employs powerful filtering capabilities using biological metadata to generate sub-networks that are much easier to explore. Through user testing and case studies, we show how PaIntDB helps researchers without a computational background to quickly identify groups of genes that are biologically relevant and formulate new hypotheses for further experiments.

2 Introduction

Pseudomonas aeruginosa is a multi-drug resistant pathogen involved in cystic fibrosis and other diseases. In the same way as all other pathogenic bacteria, the misuse and overuse of antibiotics has led to resistant phenotypes that will eventually make antibiotics obsolete. These resistance mechanisms, like any other biological process, are the result of the complex interaction between thousands of genes and gene products. Therefore, a systems-level approach, focused on groups of genes instead of individual genes, is a powerful tool to elucidate how this resistance is achieved and identify new potential drug targets. An example of this approach are differential expression experiments that test the change in expression for every gene under different experimental conditions and return a list of genes that show statistically significant changes. However, these lists are often very long (>1000 genes in experiments with *Pseudomonas*) and therefore hard to interpret by themselves.

To tackle this issue, in the Hancock lab we created PaIntDB (**P**seudomonas **a**eruginosa **Int**eractions **D**ata**B**ase), which contains 157,427 protein-protein and protein-metabolite interactions in P. aeruginosa strains PAO1 and PA14. It is a web application where a biologist can upload a list of genes and generate a network showing the interactions between them. These networks can be used to obtain insights about important genes in any given process (not only antibiotic resistance) and generate new hypotheses for further experiments. Visual exploration of the networks allows the fast identification of co-expressed gene modules and other interesting structures that would be impossible to detect by just looking at the list.

This project was started last summer by an undergraduate student who built the database by compiling interaction data from other databases and studies. I continued the project starting this April, by building a web application to generate networks using the database and a visualization module to explore them, the latter being the focus of this project for CPSC 547.

2 Related Work

PaIntDB was inspired by similar web tools developed in the Hancock Lab for systems-level biological analyses: NetworkAnalyst[1] and InnateDB[2]. These tools also generate networks by cross-referencing a list of genes with interaction databases. However, they have a much larger scope than PaIntDB, since they support multiple species and give the user more control over the visual appearance and encodings. Cytoscape[3] is a popular open-source off-line tool for biological network visualization, and allows the user to fully customize the encoding of their network attributes and layout.

Network Type	**Attributes**
BioNetwork	- Location - GO term - NodeDegree
DENetwork	- DifferentialExpression
CombinedNetwork	- Experiment

Table 1: Attributes corresponding to each network type. They are cumulative, so all the attributes in a BioNetwork are included in a DENetwork, and the same with a DENetwork and a CombinedNetwork.

All three applications have a high learning curve and unintuitive user interfaces for non-experts. PaintDB has a more specific and focused objective: allowing any Pseudomonas researcher to explore their high-throughput experiment results quickly and with minimal effort. If the user wants to customize the network's visual encoding and layout or other specific features, then PaIntDB allows downloading the networks in .graphml format that can be opened in Cytoscape or NetworkAnalyst.

A well-known issue in network visualization is the hairball problem in big, dense networks. The large amount of nodes and edges hide the interesting topological structures and it becomes impossible to extract much information from them. Matrix views have been used as an alternative to avoid this issue, since they are much more scalable without occlusion. However, users usually need training in order to extract and interpret information from matrix views, as opposed to the intuitive understanding that comes with node-link views.

3 Data and Tasks

3.1 Domain

The domain for this project is biology, specifically microbiology and systems biology.

3.2 Data

The data are undirected networks generated by the application. Every network is static after being generated but is dynamic in the sense that you will always get a different network depending on the queried genes and other user-selected parameters. Most networks have between 500 and 1500 nodes. Every node represents a protein and edges represent their biophysical interaction in the cell.

PaIntDB currently creates 3 different network types with different attributes, described below. Table 1 specifies which attributes are included in each network type. There are other attributes associated with each individual node, (such as description, accession numbers, p-values for DE genes) but they are not encoded visually and are shown in a separate table view after selecting the node(s) of interest.

- **BioNetwork**: Basic network that includes the interactions between the queried genes but no experimental information.
- **DENetwork**: DE stands for Differential Expression. Includes all the information from a BioNetwork but has additional attributes for handling differential expression experimental data.
- **CombinedNetwork**: TnSeq (Transposon Sequencing) is another recent high-throughput technique to identify genes of interest under certain experimental conditions in bacteria. Combined networks include all the information from a DENetwork but have an additional attribute indicating the experiment where the gene was identified.

3.2.1 Data Attributes

Categorical

- Localization:
  - Location of the expressed protein in the cell (cytoplasm, membrane, etc.).
  - 12 maximum levels, but depends on the queried genes.
- Type:
  - The Gene Ontology[4] (GO) initiative attempts to assign functional information to all known genes using a controlled vocabulary of hierarchichal terms, called GO terms. PaIntDB performs Fischer's test to detect which terms are statistically overrepresented in the network, and the enriched terms are used to select the genes associated with that term.
  - Levels depend on the queried genes, but usually 50-100 terms.
- Experiment:
  - Indicates if the gene was identified through RNASeq (differential expression), TnSeq, or both.
  - 3 levels.
- DifferentialExpression:
  - Indicates if a gene is turned on (up-regulated) or off (down-regulated) under the experimental conditions.
  - 2 levels.

Ordered

- NodeDegree:
  - Number of direct connections to other nodes.
  - Quantitative, sequential.
  - Range depends on the network, but usually 1 to 100.

3.3 Tasks

The tool is designed to allow biologists without a computational background to explore the results of high-throughput experiments and identify interesting groups of genes to generate hypotheses for new experiments. This high-level task can be broken down and abstracted as follows:

- Analyze, consume, discover: The user will explore the networks to find interesting network regions and nodes, and generate hypotheses about the experimental conditions and how they effect the biological process of interest.

- Search, locate: The user can find a specific gene of interest in the network and see its interactions,.

- Search, browse: The user can find genes that are up-regulated, down-regulated, located in a specific part of the cell, associated to a certain GO term, etc.

- Search, explore: The user can look at the whole network to find interesting groups of genes, either by expression or by network topology.

- Query, identify: The user can select gene(s) of interest and query the database to get all the information on that protein or metabolite, including the reference for the interaction, links to other databases, etc.

4 Proposed Solution

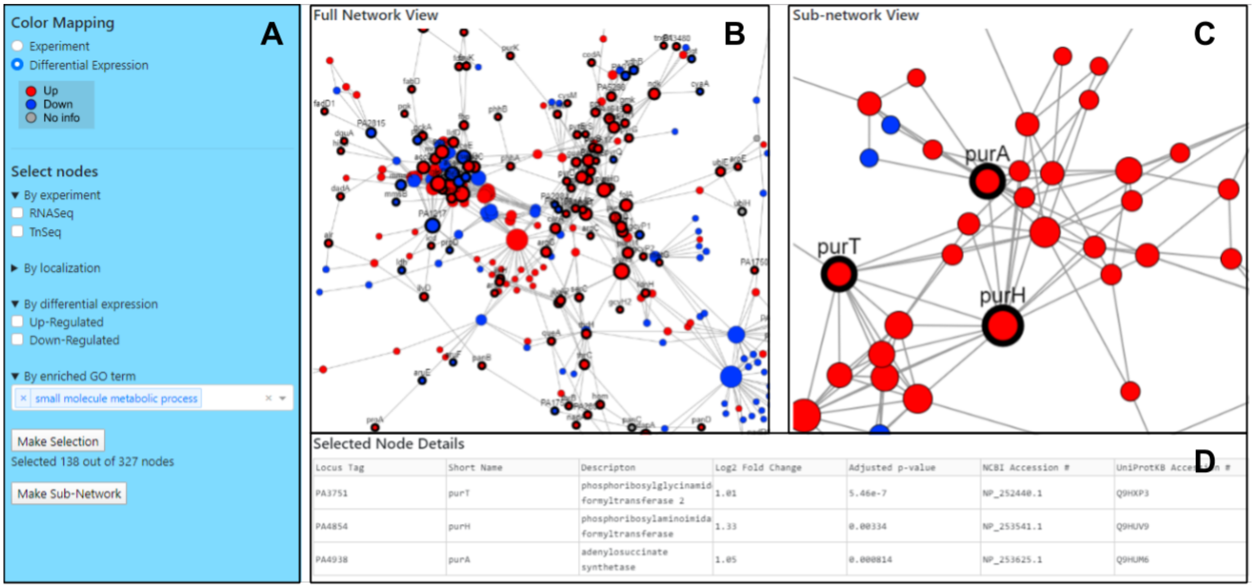
4.1 Visual Encodings

- Map node size to node degree, to identify highly connected nodes.

- Map hue to Differential Expression in DEnetworks and CombinedNetworks to identify up- and down-regulated genes.

- Map hue to Experiment in CombinedNetworks.

The two hue encodings can be toggled interactively by the user.



Color Mapping

○ Experiment

● Differential Expression

● Up
● Down
● No info

Select nodes

▼ By experiment

□ RNASeq

□ TnSeq

▶ By localization

▼ By differential expression

□ Up-Regulated

□ Down-Regulated

▼ By enriched GO term

× small molecule metabolic process

Make Selection

Selected 138 out of 327 nodes

Make Sub-Network

Selected Node Details

Locus Tag	Short Name	Descripton	Log2 Fold Change	Adjusted p-value	NCBI Accession #	UniProtKB Accession #
PA3751	purT	phosphoribosylglycinamid formyltransferase 2	1.01	5.46e-7	NP_252440.1	Q9HXP3
PA4854	purH	phosphoribosylaminoimida formyltransferase	1.33	0.00334	NP_253541.1	Q9HUV9
PA4938	purA	adenylosuccinate synthetase	1.05	0.000814	NP_253625.1	Q9HUM6

Figure 1: PaIntDB User Interface Components. A: User input panel. B: Full network view. C: Sub-network view. D: Table view.

4.1 Interface

- Left panel (Fig. 1A): user input for toggling the color mapping and filtering the main network view.
- Full network view (Fig. 1B): Shows the whole generated network. Nodes can be selected with the filters from the panel.
- Sub-network view (Fig. 1C): Shows the filtered sub-network. Nodes can be selected by clicking
- Table view (Fig. 1D): shows database information for the selected nodes.

PaIntDB has many interactive features, including geometric zooming, panning and node highlighting in the two network views. The node selection in the main view is made with node attributes to facilitate network exploration. The nodes are positioned using the Fruchterman-Reingold algorithm that models the network as a physical mass-spring system, where nodes have masses that repulse each other and edges are springs that attract connected nodes together.

As the networks get bigger, the hairball problem appears, and visual clutter hinders visual exploration. Therefore, PaIntDB allows filtering based on the node attributes to generate sub-networks from the main, whole-network view. On the main view, users can select genes based on attribute filters (location, GO term, experiment or differentialExpression) to look for specific genes in the network based on prior biological knowledge. The filtering system allows queries such as “find all up-regulated genes identified through both TnSeq and RNASeq experiments associated with DNA Repair located in the cytoplasm”. The user can then visualize a sub-network of these genes and select/de-select genes by clicking on them or with box selection. Every time a node is selected, the table view will show the node’s information from the database. Table 2 summarizes the problem solution with the what-why-how framework.

The initial idea was to have just one full network view, but test users in the Hancock lab noted that seeing hundreds of nodes isn’t that different from looking at a list with hundreds of genes. To tackle this, my first thought was to do topological community detection to get node clusters. These clusters would then be visualized as parent nodes that the user could expand/collapse. However, implementing this was harder than expected, and in bigger networks single clusters would still have hundreds of nodes. After thinking about all the biological data included in the database, I had the idea to filter the nodes based on this information and generating a smaller network using these nodes, making visual exploration a lot easier.

What: Data	Node/Link data; quantitative and categorical attributes .
Why: Tasks	Find interesting groups of genes through network exploration and retrieve database information.
How: Encode	Hue to experiment and differentialExpression (toggled by user), size to node degree.
How: Reduce	Filtering by experiment, differentialExpression, location, GO term.
How: Facet	Create sub-network using filtered nodes. Separate table view with all database information for the selected nodes.
How: Manipulate	Zoom, pan, select.

Table 2: What-why-how framework.

5 Implementation

All of PaIntDB is implemented in Python. The back-end uses the sqlite3 library to query the database, pandas to handle and filter the queried data, and networkx to create and manage the network objects. The GUI to generate the networks was made using Dash, a library to build interactive web applications in Python that abstracts all of the JavaScript, React.js, HTML and CSS code underneath. The visualization module uses the Dash Cytoscape library, which similarly abstracts the JavaScript Cytoscape.js library into easy-to-use Python components. This library is a high-level framework for network visualization, and includes interactive features such as zooming, panning and node selection out-of-the-box.

After a network is generated, it is converted to a .json file that can be used in Dash Cytoscape. A Pandas dataframe is also generated to store the node attributes in table format. This dataframe is filtered and shown in the table view depending on the user-selected nodes. The filters are created dynamically from the network attributes, so only the levels present in the network are shown in the GUI. Most of the work involved designing dynamic queries to filter the dataframe according to the user-selected filters, and ensuring they worked properly when combining multiple attributes.

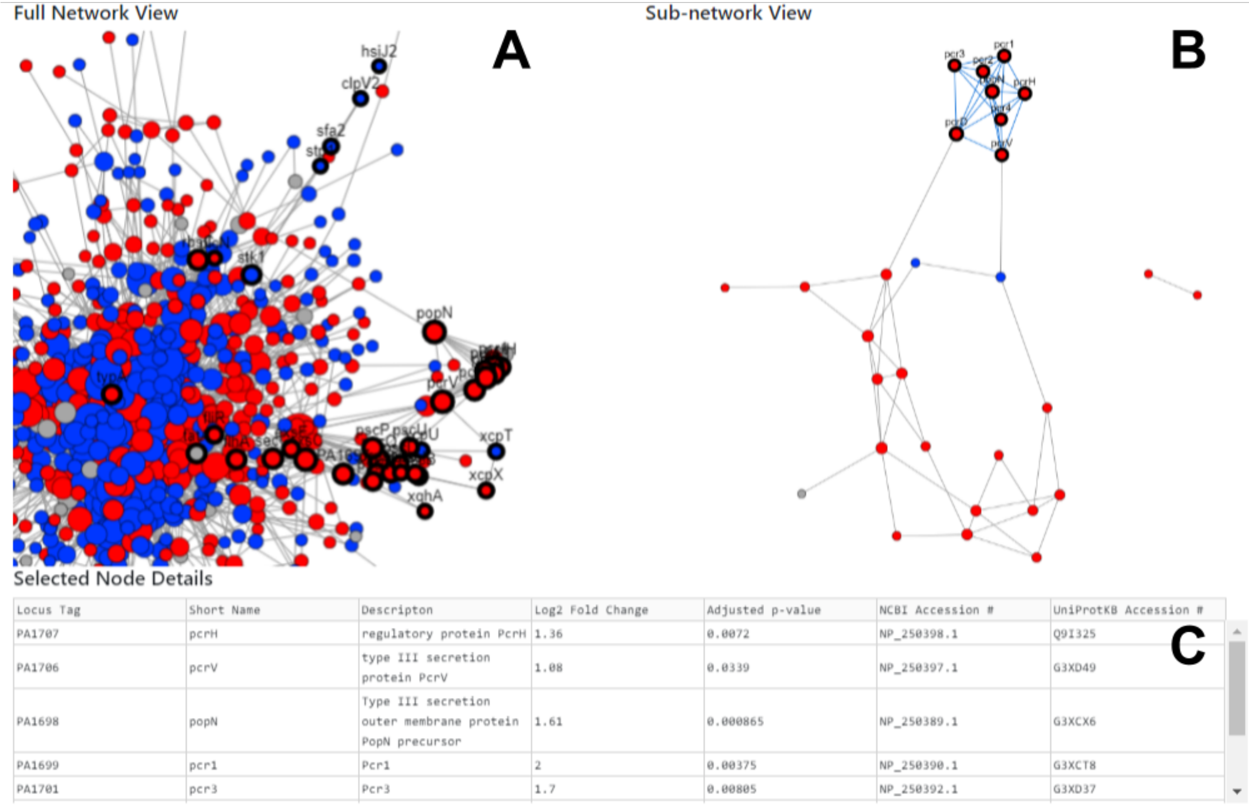
6 Results

6.1 Use Case Scenario

A microbiologist has two datasets (RNASeq and TnSeq) identifying important genes in *P. aeruginosa* treated with the antibiotic azythromycin under two different growth media. She's investigating how different media affect antibiotic resistance. After uploading her lists of genes with associated expression values, PaIntDB generates a network with 833 nodes that is essentially a hairball (Fig. 2A). She's interested in how peptide secretion changes under these conditions, so she searches for that GO term in the filters. She selects those genes and generates a sub-network with 39 nodes, much smaller than the original network. By changing the color mapping, she immediately notices that most genes are up-regulated, and identifies a cluster of highly connected genes at the top (Fig. 2B). With the information provided by the database, she finds they are part of the type III secretion system (Fig. 2C). With this in mind, she does a literature review and learns this system is like a molecular needle that detects host cells and secretes proteins to help infection. After a very short exploration, she now knows that certain growth media promote the over-expression of these genes related to infection, and can design experiments to study these genes specifically to see if there is a change in antibiotic resistance.

6.2 Test user feedback

Two microbiologists from the Hancock lab used PaIntDB to analyze their data and the application was demoed once in a lab meeting. The main complaint was interpreting the hairballs when the application just had the one full network view. After adding the second view, they agreed that the biological filters to



Selected Node Details

Locus Tag	Short Name	Descripton	Log2 Fold Change	Adjusted p-value	NCBI Accession #	UniProtKB Accession #
PA1707	pcrH	regulatory protein PcrH	1.36	0.0072	NP_250398.1	Q9I325
PA1706	pcrV	type III secretion protein PcrV	1.08	0.0339	NP_250397.1	G3XD49
PA1698	popN	Type III secretion outer membrane protein PopN precursor	1.61	0.000865	NP_250389.1	G3XCX6
PA1699	pcr1	Pcr1	2	0.00375	NP_250390.1	G3XCT8
PA1701	pcr3	Pcr3	1.7	0.00805	NP_250392.1	G3XD37

Figure 2: PaIntDB use case scenario. A: Full network is extremely hard to interpret. B: Sub-network contains less nodes and can be easily inspected. Most genes are up-regulated and there is a cluster of highly-connected nodes at the top. C: Selected genes are part of the type III secretion system.

select nodes and generate sub-networks helped them identify groups of importance. They also mentioned the obscure error messages, which is understandable since PaIntDB is still a prototype so most exceptions are caught with the default Python or library messages.

7 Discussion and Future Work

The visualization module in PaIntDB succeds in allowing biologists to quickly identify genes of interest through the combination of filtering and visual inspection of the resulting sub-networks. Using the biological metadata as filters allows biologists to use their previous knowledge to look at specific, biologically-relevant groups of genes, especially the GO term functional information.

PaIntDB still a work in progress and I will continue to work on it after the course, since it's part of my thesis project. Currently, the sub-netwokrs are created by simply removing the unselected nodes from the full network. This results in many orphaned nodes that have no direct connections to the rest of the sub-network. A better approach is to implement an algorithm that keeps the minimum number of unselected nodes from the full network that are necessary to connect the orphan nodes in the sub-network. Although the optimal solution to this problem is NP-hard, there are many algorithms that find good approximations with good performance.

PaIntDB could also take more advantage of the network topology to identify interesting structures and nodes. The only network topology statistic used so far is the simplest: node degree. Other topological information, such as node centrality and betweenness, could give a better measure of a node's importance in the network. The parameters for the layout algorithm should change depending on the network topology, since I had to tweak them manually depending on the network.

The functional GO terms have a hierarchical structure, so that filter could be improved by selecting terms at a specific level in the hierarchy, since most networks have 50-100 statistically enriched terms that are hard to organize mentally just by looking at a drop-down/search menu. Additional filters will be added to find genes related to antibiotic resistance, which is the main focus of *Pseudomonas* research.

8 Conclusion

PaIntDB is a tool that allows biologists to quickly visualize and explore results from high-throughput experiments. The generated networks, combined with filters based on prior biological knowledge are a powerful combination to get a fast, high-level understanding of how groups of genes are behaving. By selecting nodes in either view, the user can retrieve all their metadata from the database to generate new hypotheses of the genes' role under the experimental conditions. Therefore, PaIntDB succeeds in getting a systems-level understanding of biological processes by visualizing specific groups of genes of biological relevance. PaIntDB is a work in progress, and the next steps involve improving the sub-network implementation, the filters' graphical interface and the user experience.

References

1: Zhou, G., Soufan, O., Ewald, J., Hancock, R. E. W., Basu, N., & Xia, J. (2019). NetworkAnalyst 3.0: a visual analytics platform for comprehensive gene expression profiling and meta-analysis. Nucleic Acids Research, 47(W1). p. 234-241.

2: Breuer, K., Foroushani, A. K., Laird, M. R., Chen, C., Sribnaia, A., Lo, R. & Lynn, D. J. (2012). InnateDB: systems biology of innate immunity and beyond-recent updates and continuing curation. Nucleic Acids Research, 41(D1). p. 1228-1233.

3: Shannon, P., Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B & Ideker T. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. Genome Research, 13(11), 2498-2504. doi: 10.1101/gr.1239303