A Method for Connectivity Map  
Query Result Refinement and Visualization  
Theodore E. Natoli  
A Thesis in the Field of Biotechnology  
for the Degree of Master of Liberal Arts in Extension Studies  
Harvard University  
September 2014

# Abstract

The Connectivity Map (CMap) is a database of gene expression signatures obtained from experiments in which cultured human cells are treated with pharmacologic and genomic perturbagens. A typical use case of this database is for a researcher to query with a signature of a cell state of interest and use the matching perturbagens to develop a functional hypothesis for follow-up. Current pattern matching algorithms that perform CMap queries suffer from a universal weakness – the enormous size and richness of signatures in CMap means that a query typically generates hundreds of strong connections. These connections are hard to distinguish, thereby making prioritization difficult. An interconnectivity-based method of query result refinement, whereby query results that are highly interconnected amongst themselves are highlighted over singletons, proves an effective solution to the prioritization problem. To implement this method, I built a web-based tool that displays CMap query results visually in a graph layout and helps identify highly interconnected sub-groups of signatures. Using this tool and a set of curated queries, I have verified that highly interconnected query results are frequently biologically relevant and occur much more frequently than would be expected by chance alone. The tool has also proved useful in enabling hypothesis generation for novel queries.

# Introduction

## The Research Problem

The CMap database, built and maintained at The Broad Institute, is a compendium of gene expression signatures resulting from the treatment of cultured human cells with perturbations such as small molecule compounds (CP), short hairpin RNAs (shRNA), or over-expression constructs (OE). The utility of the CMap database is that of a gene expression search engine. Users are able to pose questions about relationships between cellular states and formulate hypotheses based on similarities or differences in the states’ gene expression signatures, or the sets of genes that are differentially regulated by a particular perturbation.

Hypotheses are generated by posing search queries into the database and examining the query results. A CMap query is a focused question in which a user inputs a gene expression signature, called the query, and computes the similarity, or connectivity, between his/her query and other signatures in the database. Positive connectivity indicates that two signatures’ expression changes are similar and vice versa. Researchers can use CMap to find connections between signatures within or external to the database. Hypotheses may be in the form of “the shRNA knockdown of gene X connects to shRNA knockdown signatures of pathway Y members, so X is probably a member of Y.” Or perhaps “the signature of compound Z connects to the knockdown signature of gene X, so perhaps X is the target of Z.”

Lamb et al. demonstrated a more directly therapeutically relevant use of the original incarnation of CMap when they discovered that the signature of sirolimus connected strongly to a signature of dexamethasone sensitivity. Dexamethasone is a treatment for acute lymphoblastic leukemia (ALL), but many patients eventually become resistant to it. The CMap connection between sirolimus and dexamathasone sensitivity suggested that sirolimus might be effective in reversing resistance in ALL patients who had become resistant to dexamethasone. A follow-up experiment confirmed that sirolimus conferred dexamathasone sensitivity to CEM-c1 cells, a previously dexamethasone-resistant cell line .

The CMap database contains over 400,000 signatures spanning over 70 cell types. Because of this large size, interpreting and prioritizing query results has become difficult. For example, accepting only the top one percent of connections yields nearly 4,000 signatures. Follow-up on such a large number of primary hits is nearly impossible in most cases. However, within a set of initial query results, there frequently exists a set or sets of signatures that are more tightly interconnected with themselves than with other signatures. These interconnected sets are more likely to be indicative of robust biological signal and should therefore be prioritized over other singleton connections. The goal of this work was to build a web-based tool to implement an algorithm to identify subsets of high interconnectivity within lists of initial query results and to visualize the relationships between these subsets in a graph layout. This tool is useful in refining initial CMap query results into smaller, more actionable lists of connections that can be further investigated in secondary assays.

## Gene Expression Profiling

Gene expression profiling is the simultaneous measure of the RNA transcript levels of many genes within a cell or group of cells. These measurements can help to provide insight into the cellular state or states of the cells in question. For example, if many cell-cycle genes are observed to be active, it could suggest that the cells are actively dividing. Conversely, if many apoptotic genes are active, the cells might be dying. Frequently, the goal of gene expression profiling is to identify genes that are differentially regulated between one or more sets of conditions. For example, one might measure expression in cells that have and have not been treated with a drug of interest, and then compare the resulting expression profiles to identify genes that are substantially up- or down-regulated in the treated cells relative to the untreated. Current technologies, such as the microarray, allow for many such gene expression experiments to be run in parallel, enabling the comparative analysis of hundreds or thousands of expression profiles corresponding to an equal number of experimental conditions. Similarly, expression profiling can be used to identify genes differentially regulated between disease and normal states. van’t Veer et al. used gene expression profiling to identify a set of genes that were predictive of breast cancer metastasis . Because of its ability to identify such signatures, gene expression profiling is a powerful and often-used tool in contemporary biology.

## Gene Set Enrichment Analysis (GSEA)

Gene Set Enrichment Analysis (GSEA) is an analytical approach designed to extract biological insight from gene expression data . It leverages groups of genes, called gene sets, that share some biological commonality (i.e. members of a cellular signaling pathway) and computes their enrichment, or trend towards the top or bottom, of a ranked list of genes generated by comparing expression profiles across two experimental classes (i.e. tumor vs. normal). For example, one might define many sets of genes, each corresponding to a cellular pathway. One could then rank-order all genes by their differential expression when comparing profiles of tumor vs. normal samples. Lastly, one could compute the enrichment of each pathway in the rank-ordered list to attempt to identify pathways that might be active in the particular tumor in question.

Mechanically, GSEA computes a Kolmogorov-Smirnov (KS)  
statistic when comparing a given gene set to a given ranked list . Effectively, this amounts to walking down the ranked list, increasing a running-sum statistic when one encounters a gene in the gene set and decreasing it for genes not in the gene set. The magnitude of the increment depends on the correlation of the gene with the phenotype. The enrichment score is the maximum deviation from zero encountered in the walk . GSEA has been used extensively for identifying coherent sets of genes that are collectively modulated under certain disease states and/or experimental conditions. In fact, a GSEA software suite and an accompanying online database exist to facilitate comparisons between novel and curated gene sets .

## The Connectivity Map

The Connectivity Map (CMap) is a database containing the gene expression signatures resulting from treating cultured cells with various chemical and genomic perturbations . The purpose of CMap is to serve as a lookup table of functional annotation. These annotations might be derived by comparing signatures within the CMap database or by querying the database with externally generated signatures. The database itself can be thought of as a large matrix where each row is a gene and each column is an experiment in which a particular perturbagen was profiled under a given set of conditions (i.e. cell context, dose, treatment time, etc.). The values in the matrix are differential expression measures generated by comparing the expression levels of the genes across perturbed and control states. Thus, each column of the matrix can be thought of as a given perturbagen’s expression signature.

### Computing Connections in the Connectivity Map

A primary use of the CMap database is to compare the signatures of different perturbations and to assess their similarity. Perturbagens that, when used to treat cultured cells, result in similar gene expression consequences will yield similar CMap signatures. Such signatures are said to be positively connected in the CMap sense. Conversely, perturbagens that elicit inversely related expression consequences are said to be negatively connected. For a given query signature Q and a reference signature R, the weighted connectivity score (WTCS) is computed by computing and integrating two KS statistics, one each for the n most up- and down-regulated genes in Q. The algorithm proceeds as follows:

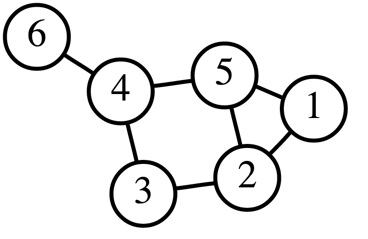
1. Order Q
2. compute ESup as the enrichment of the n most up-regulated genes in R
3. compute ESdn as the enrichment of the n most down-regulated genes in R
4. compute WTCS as
   1. 0 if ESup and ESdn share the same sign
   2. $ \frac{|ES\textsubscript{up}| + |ES\textsubscript{dn}|}{2} $, where the resulting WTCS is given the sign of ESup if ESup and ESup are of different signs.

WTCS will be positive for signatures that are positively related and negative for those that are inversely related.

A common CMap use case is to select a given query signature Q from the database and compute its similarity to all other signatures. The remaining signatures can be ranked according to their connection strength with Q. The connections can be used to gain insight and form hypotheses about Q. For example, if Q is a signature of a novel compound and it connects strongly to signatures of compounds of a known pharmacological class, one might hypothesize that the novel compound is also a member of this class. Similarly, if Q connects strongly to the knockdown signature of gene G, one might hypothesize that G is the novel compound’s target. Q need not be a signature from the CMap database. For instance, it might be the signature of some disease and one might seek connections to genes whose modulation could be causing the disease or to compounds that could have therapeutic relevance.

## Graphical Depictions of Biological Phenomena

In the fields of computer science and mathematics, a graph is a means by which to represent a set of objects and the relationships between them. It is frequently depicted as a set of nodes, where each node represents an object. Connections, where they exist, are represented by edges (Figure  [fig:graph]).

 [fig:graph]

Although they originated in other fields, graphs have frequently been used as tools to model biological phenomena. Graphical models of protein interaction networks, gene expression networks, and other similar phenomena are commonplace. Friedman used graphical models to infer and visualize gene regulatory networks . Lage et al. used graphical models to characterize existing and elucidate novel protein-protein interaction networks . Because of its widespread use and adoption, the graphical model is an appropriate, familiar, and effective means to depict connectivity between CMap signatures. In the graph visualization generated by the application, CMap signatures are represented as nodes. Where a connection exists between to signatures, a line is drawn connecting their nodes. Thus, the user is able to easily see which signatures are connected to each other and can easily identify highly interconnected subsets of signatures, if they exist.

# Methods

## Computing Connections and Visualizing Connections

Connections between CMap signatures were computed using WTCS.

then use rankpoint to convert to percentile rank

In order to facilitate application performance, these connections were all be pre-computed and stored in a database. This way, the application is able to simply look up connectivity scores instead of computing them on-the-fly.

Where the connection strength between two nodes exceeds a given threshold, the application asserts that a connection exists between the signatures, a line is drawn between them. The application usse

Users interact with the application by inputting a list of signatures L that have resulted from running a CMap query.

also possible to input an arbitrary list. maybe mention this in discussion

describe how the app actually works

## Software Components

### Front End: HTML & D3.js

Hypertext markup language (HTML) is and has been the standard language for displaying information over the Internet within web browsers. HTML5, the most recent revision of the HTML standard, will be used as the framework of this application. HTML5 offers many useful features for application development and is supported by most modern web browsers .

To support user interaction, the graph visualization is built using D3.js, a JavaScript library for data visualization. Created by Mozilla in 1995, JavaScript is a programming language that is interpreted by most modern web browsers and allows developers to create interactive elements within web pages . D3, short for Data Driven Documents, is a JavaScript library written by Mike Bostock and designed specifically to enable rich and interactive data visualizations . D3 is particularly well-suited for visualizing CMap connections because of its ability to easily integrate and bind data to on-screen elements. It has been used in many similar projects and is capable of generating the types of visualizations this application requires.

### Back End: Node.js & MongoDB

MongoDB is a database system developed by MongoDB Inc. Unlike traditional Structured Query Language (SQL) database systems that require rigid data storage schema, MongoDB’s schema is very loose and fluid. Rather than storing data in tables that may or may not be linked to each other, Mongo stores data in “collections,” where each collection is simply a list of “documents.” Documents are simply data objects that can have any number of attributes and each document need not have the same attributes as others . Perhaps the main benefit of using MongoDB is that it natively stores data in JavaScript Object Notation (JSON) format . JSON is the data object format used by JavaScript, so using MongoDB to store the connectivity data means that when the application queries MongoDB, the database will respond with data in a format the application can easily handle.

MongoDB documents are simply JSON objects. These objects contain key-value pairs and the values can be accessed by providing the appropriate keys, or attributes. A CMap connection can be modeled as a very simple JSON object with the following attributes:

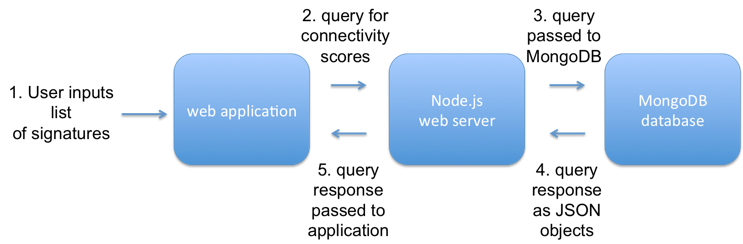
1. perturbation 1
2. perturbation 2
3. score

An example CMap connection stored as a JSON document might look like this:

{  
 "perturbation\_1": "vorinostat",  
 "perturbation\_2": "trichostatin-a",  
 "score": 0.98  
}

In the example above, vorinostat and trichostatin-a have a connection strength of 0.98. Providing the names of the two perturbation is enough to uniquely identify this and any CMap connection. MongoDB allows for searching over the values of all documents that contain a given key or set of keys. The application stores each CMap connection as a document in a single MongoDB collection. Based on the user’s input set of query results (perturbation names), MongoDB is able to retrieve all connections between the query results by looking up all documents where the perturbation\_1 and perturbation\_2 fields are members of the input query result set and then return the results to the application as a JSON object.

Node.js is a JavaScript-based platform for web server development. It implements an event-driven paradigm, which means that it enables writing programs built for quickly responding to inputs from a user or another application . In this project, Node.js acts as the web server that handles requests from the web application and query responses from MongoDB. It acts as the middle layer that shuffles data between MongoDB, where it is stored, and the web application, where it is displayed. Figure  [fig:appdataflow] illustrates how data flows through the various front and back end layers of the web application. Node.js is appealing for this use case because, like MongoDB and D3, it is based in JavaScript. It therefore allows for easily passing query parameters from the application to MongoDB and query results as JSON objects from MongoDB to the application.

 [fig:appdataflow]

# Results

## Highly Interconnected Query Results are Frequently Biologically Relevant

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## Application Efficiently Identifies Interconnected Results

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## Novel Query Results That Show Interconnectivity

could be useful to talk about ERG here

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Summary and Conclusions

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Appendix A: Source Code

## CSS

## HTML

## JavaScript

## R