Visualizing and Refining Connectivity Map Query Results

December 3, 2013

Proposal for a
Thesis in the Field of
Biotechnology

In Partial Fulfillment of the Requirements
for a Master of Liberal Arts Degree

Harvard University
Extension School

Theodore Natoli
51 Lewis Avenue, Apartment 3
Arlington, MA 02472
857-498-1946
tnatoli@fas.harvard.edu

Proposed Start Date: December 3, 2013
Anticipated Date of Graduation: December 3, 2013
Thesis Director: Aravind Subramanian

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. I hypothesize that one mode of prioritization is to highlight query results that are highly interconnected amongst themselves over singletons. The goal of this work is to provide a web-based tool for implementing an interconnectivity-based method of query result refinement and for visualizing CMap query results in a graph layout.

1 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 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.

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 (Tissing, Meijerink, den Boer, & Pieters, 2003). 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 (Lamb et al., 2006).

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. I propose that within a set of initial query results, there will frequently exist 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 is to build a web-based tool to implement an algorithm ~~to~~ identif~~y~~ subsets of high interconnectivity with lists of initial query results and to visualize the relationships between these subsets in a graph layout. I propose that this tool will be useful in refining initial CMap query results into smaller, more actionable lists of connections that can be further investigated in secondary assays.

2 Key Terms

1. Gene expression profiling: the parallelized collection of many simultaneous gene expression measreuments

2. Gene set enrichment analysis (GSEA): an approach used to test for a gene set's overall trend toward~~s~~ the extremes of a ranked-list of genes.

3. Kolmogorov-Smirnov (KS) statistic: a statistical measure used to quantify the trending of one group's members toward~~s~~ the extremes of another group's distribution.

4. ~~g~~raphical model: a computational tool for approximating the interre-

lationships between a set of objects.

5. Connectivity Map (CMap): a database of gene expression signatures spanning many treatment types and cell contexts.

6. CMap connection: a quantified similar or dissimilar relationship between two gene expression signatures.

3 Background

3.1 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 technology, 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 (van 't Veer et al., 2002). Because of its ability to identify such signatures, gene expression profiling is a powerful and often-used tool in contemporary biology.

3.2 Gene Set Enrichment Analysis (GSEA)

Gene Set Enrichment Analysis (GSEA) is an analytical approach designed to extract biological insight from gene expression data (Subramanian et al., 2005). 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 toward~~s~~ 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 (Hollander

& Wolfe, 1975). 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 random walk (Subramanian et al., 2005). GSEA has been used extensively for identifying coherent sets of genes that are collectively modulated under certain disease 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. Cite MsigDB and GSEA.

3.3 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 (Lamb et al., 2006). 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.

3.3.1 Computing Connections in the Connectivity Map

A primary use of the CMap database is to compare the signatures of different perturbations and 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 ES_{up} as the enrichment of the n most up-regulated genes in R
3. compute ES_{dn} as the enrichment of the n most down-regulated genes in R
4. compute WTCS as

    (a) 0 if ES_{up} and ES_{dn} share the same sign

(b) $(|ES_{\text{up}}| + |ES_{\text{dn}}|2), wheretheresultingWTCSisgiventhesignofES_{\text{up}}$

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.

3.4 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 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 1).

Although they originated in other fields, graphs have frequently been used as tools to model biological phenomena. Graphical models of protein

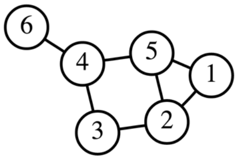

Figure 1: An example of a graph. The six nodes are connected by seven edges. All nodes other than node 6 have at least two edges.

interaction networks, gene expression networks, and other similar phenomena are commonplace. Friedman used graphical models to infer and visualize gene regulatory networks (Friedman, 2004). Lage et al used graphical models to characterize existing and elucidate novel protein-protein interaction networks (Lage et al., 2007). Because of its widespread use and adoption, the graphical model is an appropriate, familiar, and effective means to depict connectivity between CMap signatures.

3.4.1 Proposed use of Graphs in This Work

In this work, a graph will be used to depict the existence and strength of connections between gene expression signatures in the CMap database. The graph lends itself very well to this use case, as signatures will be represented as nodes and their pairwise connections by the edges. Edges will only be depicted between nodes with a non-zero connectivity score.

4 Methods

4.1 Computing Connections and Visualizing Connections

Connections between CMap signatures will be computed using WTCS. In order to facilitate application performance, these connections will all be precomputed and stored in a database. This way, the application can simply look up connectivity scores instead of computing them on-the-fly.

Users will interact with the application by inputting a list of signatures L that have resulted from running a CMap query. The application will sort L by the strength of connection to the user's query and then look up the connectivity scores between all pairwise combinations between the first N signatures within L, where N is a user-provided parameter. The application will then identify highly interconnected subsets of signatures by clustering the N signatures in pairwise connectivity space using the k-means clustering algorithm(Lloyd, 1982). Additionally, it will compute the optimal number of clusters using the GAP statistic and will report the within cluster sum of squares (WCSS) for each cluster. The WCSS is a measure of cluster tightness and could be used for cluster prioritization.

The N signatures will be displayed as a graph, where each signature is a node. Node pairs with non-zero connectivity scores will have edges drawn between them and nodes will be colored according to their cluster membership. This will allow users to easily visualize cluster concordance and relationships between clusters. The application will allow users to mouse-over nodes and see additional information about the signatures, such as the experimental parameters under which they were generated. The user will also be able to select a cluster or other user-defined group of nodes and see a word cloud generated from their annotations. The word cloud might give insight to pathway or pharmacological class membership for the nodes in question.

Finally, the application will support export of the clusters or user-defined groups of nodes into a text file for download. Figure 2 depicts a mockup of what the final application might look like.

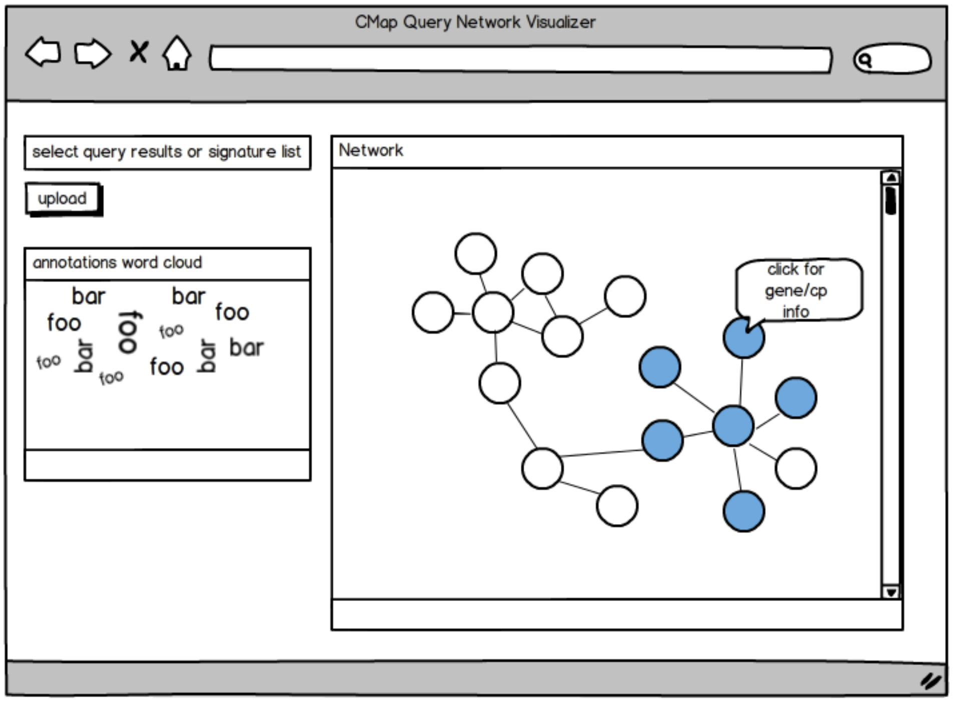

Figure 2: The App

4.2 Software Components

4.2.1 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 (W3C, 2011).

To support user interaction, the graph and word cloud visualizations will be 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 (Mozilla, 2013). D3, short for Data Driven Documents, is a JavaScript library written by Mike Bostock and designed specifically to enable rich and interactive data visualizations (Bostock, 2013). 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 my application will require. Figure 3 and Figure 4 illustrate examples of D3 being used to generate interactive graph and word cloud visualizations.

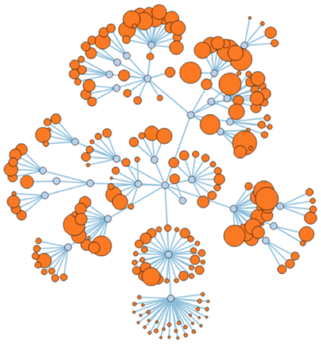

Figure 3: A graph generated using D3.js

4.2.2 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 (MongoDB, 2013). Perhaps the main benefit of using MongoDB is that it natively stores data in JavaScript Object Notation (JSON) format (ECMA, 1999). 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 Schema

Figure 4: A word cloud generated using D3.js

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. signature 1

2. signature 2

3. WTCS

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

```
{
        "signature1": "signature_1_ID",
        "signature2": "signature_2_ID",
        "wtcs": 0.65
}
```

Providing the identifiers of signatures 1 and 2 are 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. I propose to store each CMap connection as a document in a single MongoDB collection. Based on the user's input set of query results (signature IDs), MongoDB will easily be able to retrieve all connections between the query results by looking up all documents where the signature1 and signature2 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 (Node, 2013). In this project, Node.js will act as the web server that handles requests from the web application and query responses from MongoDB. It will effectively act as the middle layer that shuffles data between

MongoDB, where it is stored, and the web application, where it is displayed. Figure 5 illustrates how data will flow 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.

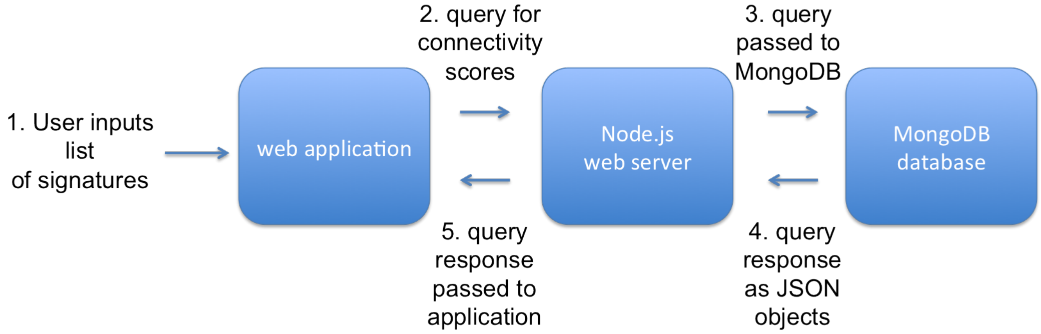
1. User inputs list of signatures
web application
2. query for connectivity scores
Node.js web server
3. query passed to MongoDB
MongoDB database
4. query response as JSON objects
5. query response passed to application

Figure 5: Application data flow diagram. The application receives a list of signatures from the user. It then sends a query for these signatures' connectivity scores to Node.js. Node.js receives the query, passes it to MongoDB, and waits for the response. Once the JSON response is received, Node.js passes it back to the application for visualization.

5 Potential Challenges

5.1 Storing Entire Connectivity Matrix in MongoDB

The CMap database contains over 400,000 signatures. Thus, the matrix of all pairwise connectivity values would be 400,000 x 400,000 or 160 billion unique values. Such a matrix would translate to a MongoDB collection of 160 billion documents. A collection of that size may prove time-consuming to query and could result in substantial lag times experienced by the application end-user. There are several options for mitigating this issue, should it arise. The first is to use only a subset of the matrix. CMap provides a high or low-quality label to each of their signatures. There are roughly 240,000 high-quality signatures, so using only those would reduce the size of the MongoDB collection to 62.5 billion. Another alternative is to use an abstracted view of the signatures. The CMap group is developing a summarized set of perturbation-centric signatures where connections between perturbations are collapsed across cell lines. The current summarized space is approximately 8,000, which would yield a MongoDB collection of 64 million, roughly thousand-fold reduction in database size. An advantageous byproduct of using the summarized space is that it would abstract away the notion of cell context and might make connectivity results easier to interpret for the end-user.

5.2 Displaying Large Graphs in a Web Browser

There may be an upper limit to the number of nodes that can be displayed within a modern web browser while still allowing for reasonable application performance. This number may vary depending on the capabilities of the end-user's computer, but testing across a few computers, each with different specifications, should yield a rough estimate of a reasonable graph size. It may be problematic if this size is well below the number of highly interconnected signatures within a given query result. If this were the case, the application would not be able to display all meaningful nodes simultaneously. Options for mitigating this issue include implementing a pan or zoom feature that would allow users to see only portions of the entire graph at a time or computationally collapsing nodes to reduce the overall number to a more manageable size. Both options could potentially increase application development time and computational burden on the end-user's machine and will need to be considered carefully if not all meaningful connections can be displayed simultaneously.

5.3 Incorporation of Signature Metadata

Each CMap signature is accompanied by a set of metadata that describe how the signature was generated. This information includes the perturbation profiled and other experimental parameters such as treatment time, dose, and cell line. It is ~~these~~ information that will be necessary for generating the word cloud visualization and other mouse-over interactions in the application. The CMap group has made this data available through an Application Programmer Interface (API), but accessing this means that my application will need to interact with a second database. Such a configuration is certainly possible, but could potentially increase lag time experienced by the end-user. If the lag time is prohibitive, it might be necessary to forego the metadata-based components of the visualization.

6 Preliminary Timeline

2013-12-15 Receive approval of proposal

2013-12-30 Investigate and validate the notion of refining hit lists base on their interconnectivity

2014-01-07 Implement graph visualizer in D3

2014-01-14 Implement node selection and highlighting

2014-01-21 Implement Node.js and MongoDB backend

2014-02-14 Implement user-based input, uploading text files and CMap query results

2014-03-01 Implement word-cloud generation based on selected graph nodes

2014-03-28 Implement filtering, showing/hiding graph nodes, brushing to select nodes based on connectivity score

2014-04-15 Implement text or spreadsheet result export

2014-05-01 Implement linking to external sites (GeneBank, ChemBank, for clicking on graph nodes)

2014-06-01 Initial draft of thesis submitted

2014-07-01 Final draft of thesis approved

2014-07-14 Bound copies submitted to Extension School

2014-11-01 Graduation

References

Bostock, M. (2013, November). D3: Data driven documents. Retrieved from `http://d3js.org`

ECMA. (1999, December). Json. Retrieved from `http://www.json.org`

Friedman, N. (2004, February). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799–805. Retrieved 2013-11-29, from `http://www.sciencemag.org/content/303/5659/799` (PMID: 14764868) doi: 10.1126/science.1094068

Hollander, M., & Wolfe, D. A. (1975). Nonparametric statistical methods. *Biometrische Zeitschrift*, *17*(8), 526–526. Retrieved from `http://dx.doi.org/10.1002/bimj.19750170808` doi: 10.1002/bimj.19750170808

Lage, K., Karlberg, E. O., Strling, Z. M., lason, P. ., Pedersen, A. G., Rigina, O., . . . Brunak, S. (2007, March). A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature Biotechnology*, *25*(3), 309–316. Retrieved 2013-11-29, from `http://www.nature.com.ezp-prod1.hul.harvard.edu/nbt/journal/v25/n3/full/nbt12` doi: 10.1038/nbt1295

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., . . . Golub, T. R. (2006, September). The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science*, *313*(5795), 1929–1935. Retrieved 2013-10-18, from http://www.sciencemag.org/content/313/5795/1929 (PMID: 17008526) doi: 10.1126/science.1132939

Lloyd, S. P. (1982, March). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *IT-28*. Retrieved 2013-11-29, from http://www.cs.nyu.edu/ roweis/csc2515-2006/readings/lloyd57.pdf doi: 10.1109/TIT.1982.1056489

MongoDB. (2013, November). Mongodb. Retrieved from http://www.mongodb.org

Mozilla. (2013, November). Javascript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

Node. (2013, December). Node.js. Retrieved from http://www.nodejs.org

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., ... Mesirov, J. P. (2005, October). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(43), 15545–15550. Retrieved 2013-10-18, from http://www.pnas.org/content/102/43/15545 (PMID: 16199517) doi: 10.1073/pnas.0506580102

Tissing, W. J. E., Meijerink, J. P. P., den Boer, M. L., & Pieters, R. (2003, January). Molecular determinants of glucocorticoid sensitivity and resistance in acute lymphoblastic leukemia. *Leukemia*, *17*(1), 17–25.

(PMID: 12529655) doi: 10.1038/sj.leu.2402733

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., ... Friend, S. H. (2002, January). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*(6871), 530–536. Retrieved 2013-11-10, from `http://www.nature.com/nature/journal/v415/n6871/abs/415530a.html` doi: 10.1038/415530a

W3C. (2011, April). Html5 differences from html4. Retrieved from `http://www.w3.org/TR/2011/WD-html5-diff-20110405/`