Molecular Re-Classification of Renal Disease Using Approximate Graph Matching, Clustering and Pattern Mining

Ramakrishna Varadarajan[1], Felix Eichinger[2], Jignesh Patel[1] and Matthias Kretzler[2].

[1] University of Wisconsin-Madison, Madison, WI and [2] University of Michigan, Ann Arbor, MI.

Abstract

Classification of patients with a chronic disease course, such as kidney diseases, uses mainly descriptive disease definitions. To develop molecular based disease stratification, we aimed to define patient subgroups by conserved transcriptional networks. Defining similarity of patients on a regulatory network level, rather than on an individual gene level, might yield more robust indicators of function. Network nodes for each patient were derived from Affymetrix microarrays of kidney biopsies compared to healthy controls. Subsequently, relations between the nodes were established by natural language processing of PubMed abstracts and automated promoter analysis for transcription factor binding sites. The resulting networks are typically noisy or incomplete in nature; therefore network similarities are determined through an approximate graph-matching tool, allowing a degree of mismatching (within a preset threshold) in the displayed transcriptional networks. Based on a similarity score the patient networks are clustered - with the goal of attaining high intra-cluster similarity (networks within a cluster are highly similar) and low inter-cluster similarity (networks from different clusters are dissimilar). To extract underlying biological mechanism inside each cluster, we employ graph mining techniques and search for frequently occurring motifs (recurring subnetworks) within each cluster, indicative of characteristic disease processes (commonly occurring phenomenon within each cluster). Motifs across each cluster are compared to define mechanistic similarities and differences between network clusters. Finally, both clusters and motifs are matched back to the established descriptive clinical classifications to compare molecular and clinical classification.

Introduction

Two Approaches:

1) **Network-based Approach:**

For each patient, create a network with genes as nodes and additional information (PPI, literature search) as edges.

Run approximate graph matching algorithm (TALE) to determine which networks are similar.

2) **Annotation-based Approach:**

Group genes by annotation and cluster patients for each of those groups.

In this poster – we focus on the networks approach

Why Networks ?

- Cross-reference information of gene lists with independent knowledge
  - ✓ Don't compare only identities, but also structures.
  - ✓ *Can* help stabilizing.
  - ✓ Will also introduce bias.
  - ✓ Since we compare individual patients (n=1), the potential profit is estimated higher then the loss.

Gene Selection

How to do that for each patient ?

No significance - revert to fold-change to make a binary decision if gene is "differently expressed".

Compare to controls:

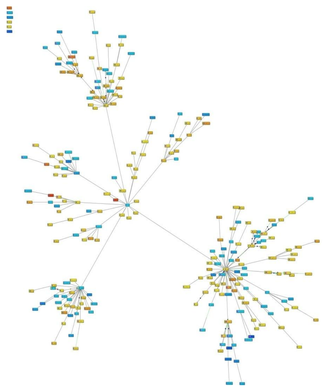
For each gene, calculate median and standard deviation in the controls.
Subtract medians of controls from patients expression values.
Result - genes with little change will have values close to 0.
If value is smaller than 2 x standard deviation, then discard gene.

Result: Gene list for each patient that differ in length and composition.

Construct Networks

Feed those ~250 gene lists into Bibliosphere.

- ➢ Generate gene networks from pub med abstracts.
  - ❖ Edges are co-citations if genes in abstracts.
  - ❖ Level of expression does not play ANY role.
  - ❖ All networks are created on the same knowledge base => subnets of the same core.

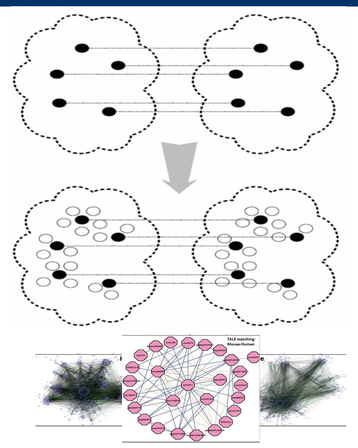


Merge Networks

• For all combinations of the 250 networks

- ❑ Perform approximate graph matching (using TALE)
  - ✓This again works solely on structure, the expression levels play no role.
  - ✓Approximate matching helps to account for noise and redundancy

Result: Pair-wise similarity of networks.



Pair-wise Network Similarity Computation

We load all graphs into the database, and use TALE to query each of the 250 networks against the database. So, basically, there are 250 X 250 comparisons and we get the pair-wise matching results.

We consider both:

- ➢ Size of the match (the number of matching nodes) and
- ➢ How similar the connectivity of nodes is in the match.

The similarity scores are computed after the shared network between the graphs is computed. To be precise, we use the following measure to access the quality of the match:

$$S_\lambda(G_1, G_2) = |\lambda| - \frac{StructDist_\lambda(G_1, G_2)}{\frac{|\lambda| \times (|\lambda| - 1)|)}{2}} \times \log |G1|$$

Under this similarity model, a higher score means more similar. Note that this similarity score is asymmetric. Therefore, for each pair of maps, we use the maximum of the two as the similarity score between the two maps. *StructDist* is the summation of the shortest distances between every matching pair of nodes in the two networks.

Clustering

A cluster is an aggregation of networks, that share some similarity. The goal of clustering is to maximize intra-cluster similarity and minimize inter-cluster similarity.

Goal: Group patients by network similarity thresholds.

Key problem → Find appropriate parameters:

- ✓ Reasonable # of members per cluster.
- ✓ Most/all patients are present in any cluster.

Clustering Algorithm

We use MCL algorithm for clustering networks, based on the pair-wise network similarities. The **MCL algorithm** is short for the **Markov Cluster Algorithm**, a fast and scalable unsupervised cluster algorithm for networks (also known as *graphs*) based on simulation of (stochastic) flow in graphs.

Before clustering the networks, we use a similarity threshold to eliminate some insignificant pair-wise network similarities. Different network similarity give different cluster results. A higher threshold would result in many smaller clusters and vice versa.

Pattern Mining in Clusters (find common motifs in clusters)

Patterns are frequently occurring sub-graphs within the networks present in a cluster. Note that, we currently only mine patterns within each cluster. This means, we have a set of patterns for each cluster.

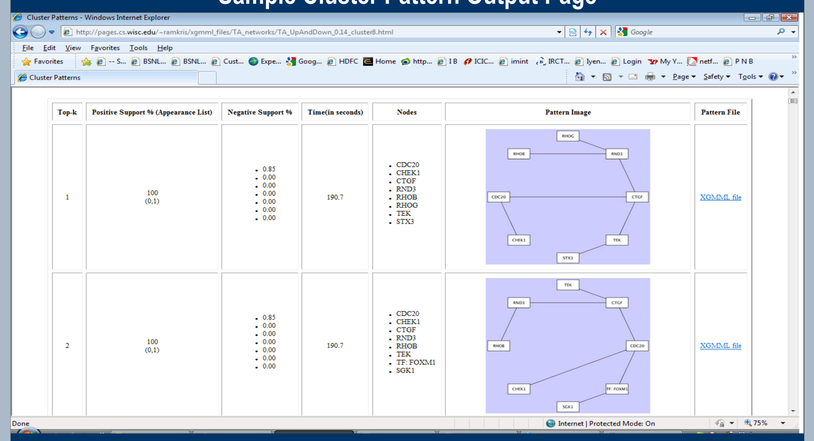
❑ In each of the clusters:

- ➢ Find common sub-networks (motifs).
- ➢ Could be used for patient classification.
- ➢ They might be a starting point to define function specific to a patient group.

In this paper, we are particularly interested in mining contrast patterns in each cluster. Contrast patterns are those with high frequency in one cluster and low frequencies in the remaining clusters. Contrast patterns are unique to a cluster and hence are particularly interesting.

Frequent pattern mining has attracted a lot of interest recently. *Frequent substructures are very basic patterns that can be* discovered in a collection of graphs. Recent studies have developed several frequent substructure-mining methods.

Sample Cluster Pattern Output Page



Summary

1. Select genes for each patient.
2. Generate network for each patient.
3. Merge networks using TALE.
4. Cluster networks using similarity determined by TALE.
5. Within each of the clusters: search for common motifs (sub-networks).