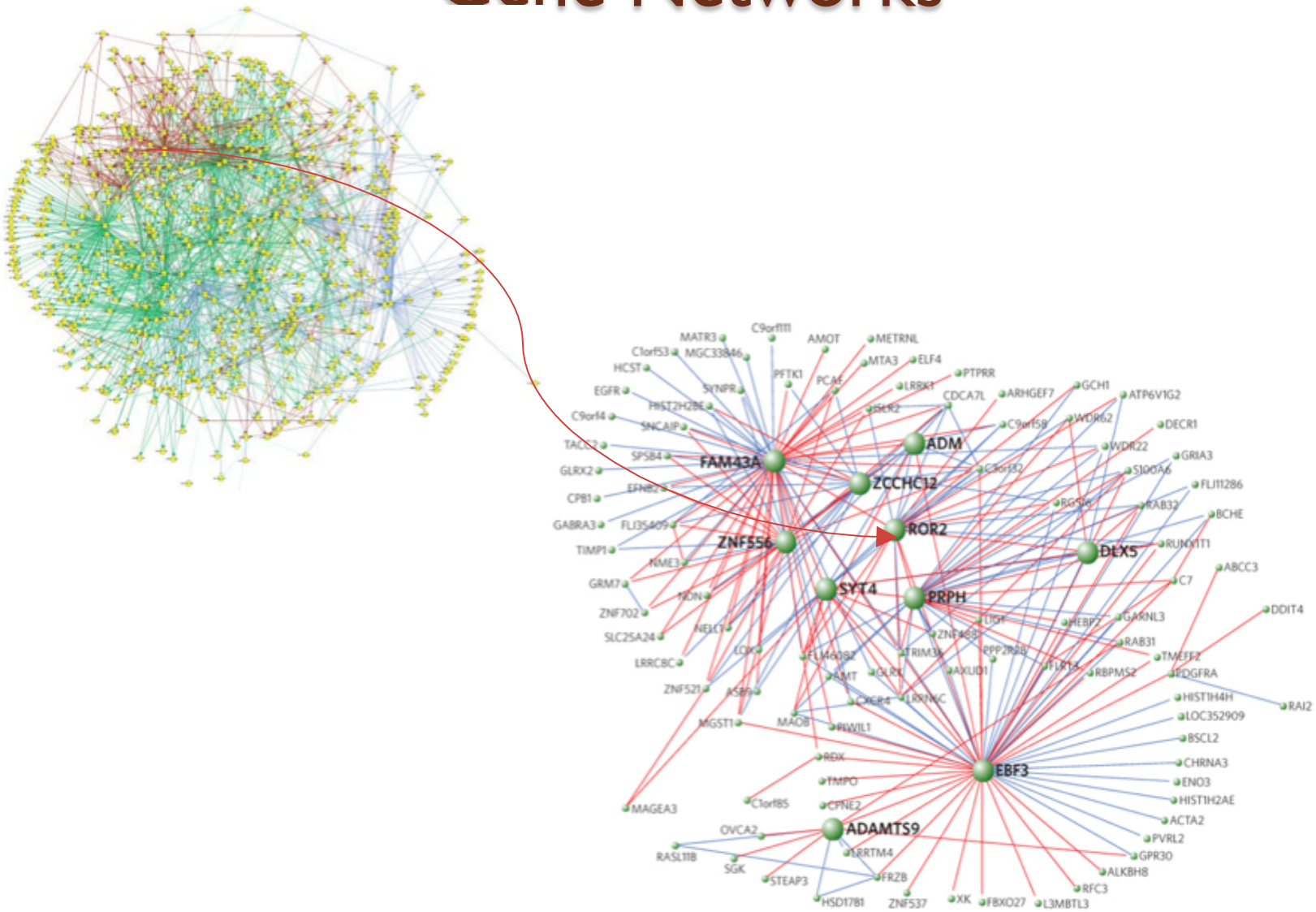
Assessing a Nonlinear Dimensionality Reduction-Based Approach to Biological Network Reconstruction

Vinodh N. Rajapakse (vinodh@math.umd.edu)
Advisor: Wojciech Czaja (wojtek@math.umd.edu)

Presentation Outline

- Problem Overview
- Solution Approach
- Implementation Notes
- Validation and Testing
- Project Schedule and Milestones
- Deliverables
- Questions and Comments

Gene Networks



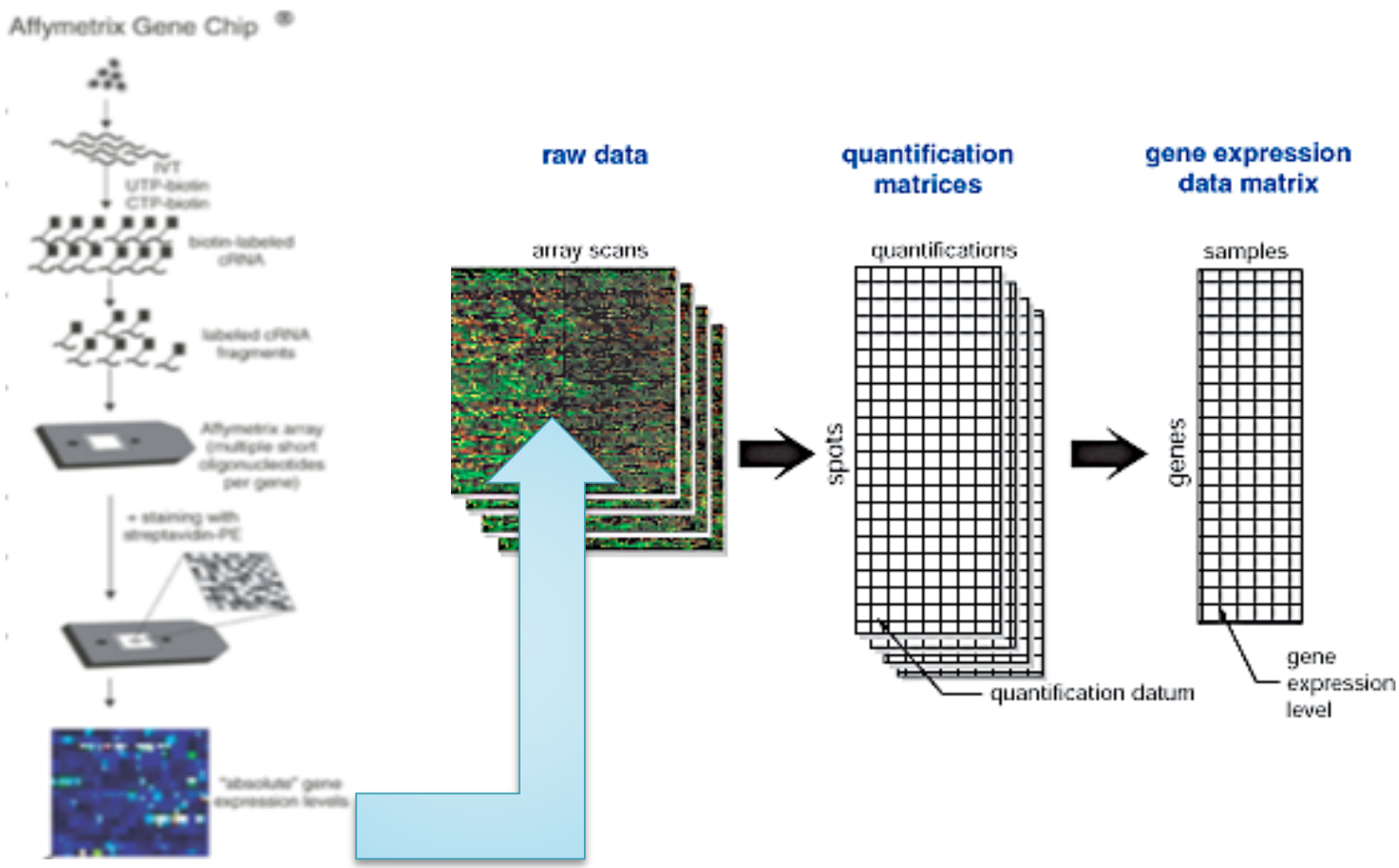
Why Build Gene (etc.) Networks?

- Gain a broader, systems level view of biological processes and their underlying functional elements
  - Avoid a narrow focus on a limited subset of driving elements
  - Incisively identify the most promising targets for experimental exploration (to derive focused data for iteratively refining models)

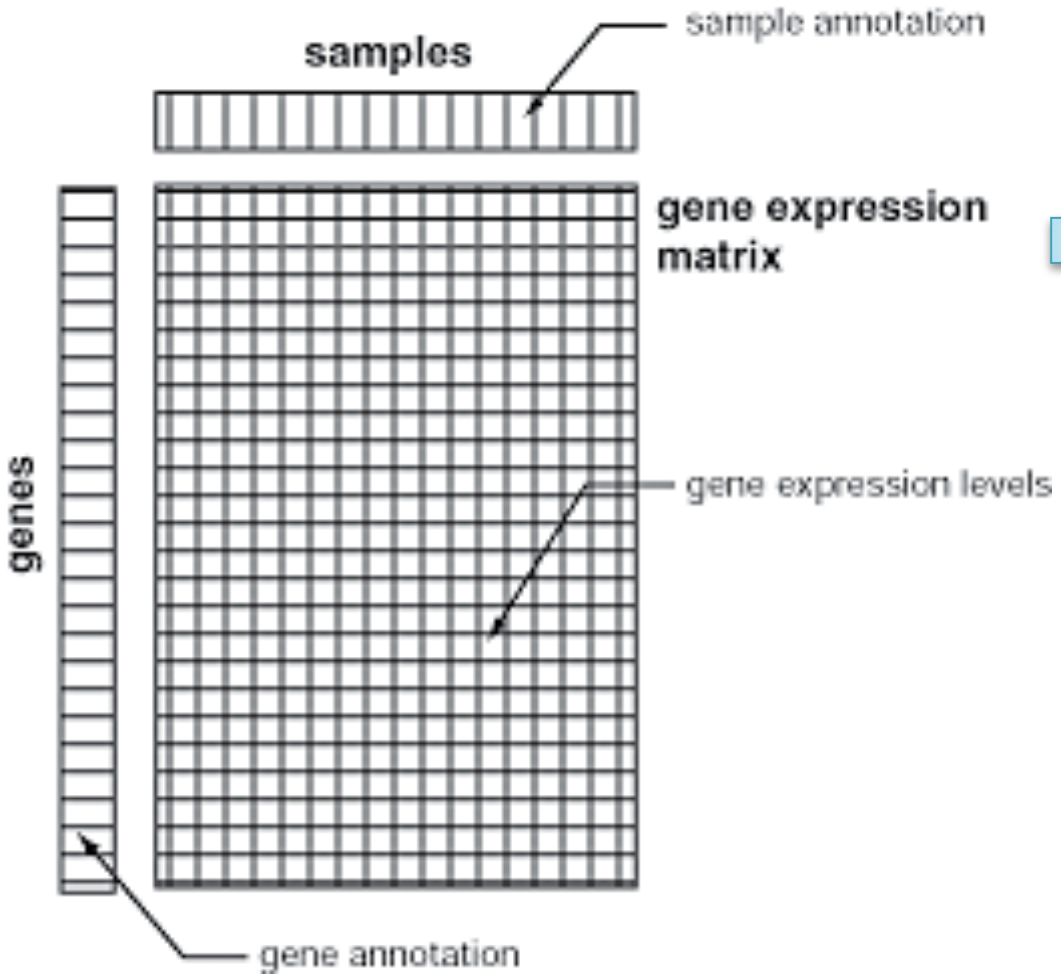
How to Build Biological Networks?

- Manually - using expert knowledge, detailed review of research results
  - Only avenue until relatively recently
  - Necessarily small scale – a few reliable (experimentally verified) nodes and links, many, many missing ones.
- Computationally – using large scale measurements of molecular expression (abundance) values over many biological samples

Gene Expression Microarrays



Starting Point: Gene Expression Data Matrix



X **(N x D)**

N = ~10000 genes
D = ~250 samples

Basic Network Construction Workflow

- Starting from from the N x D gene expression data matrix **X**, derive an N x d matrix **Y**, (d < D) using Laplacian Eigenmaps (or another dimensionality reduction technique).
- Construct an N x N matrix **W*** capturing pairwise Euclidean distances between row vectors of **Y** ('reduced gene profiles').
- Apply a statistical significance-based threshold to the elements of **W*** to obtain a network (adjacency matrix) representation.

Laplacian Eigenmaps

- Input: **X** (N x D) → Output: **Y** (N x d)
  - Let **x** = ($x_1, x_2, \ldots, x_D$) denote a row of **X**
  - Let **y** = ($y_1, y_2, \ldots, y_d$) denote a row of **Y**
- Step 1: Model Data Point Relationships
  - Build a graph G, with nodes i and j connected if $\mathbf{x}_i$ is one of the k nearest neighbors of $\mathbf{x}_j$ or vice versa (Euclidean distances used, alternatives are possible)
  - k is a local structure resolution parameter

Laplacian Eigenmaps

- Step 2: Form Weight Matrix
  - Form a diffusion weight matrix W, with entry $W_{i,j} = \exp\{ - \| x_i - x_j \|^2 \}$, if i and j are connected; 0 otherwise.
- Step 3: Solve Minimization Problem

$$\min_{(Y^T DY = I)} \frac{1}{2} \sum_{i,j} \| y_i - y_j \|^2 W_{i,j}$$

Laplacian Eigenmaps

- Step 3 (cont.) Solve Eigenvalue Problem
  - Given weight matrix **W**, let **D** be a N x N diagonal ('connectivity') matrix with entries recording the sum of edge weights for each data point-derived node
  - Let **L** = **D** – **W** denote the Laplacian matrix
  - We have:

$$\min_{(Y^T DY=I)} \frac{1}{2} \sum_{i,j} \| y_i - y_j \|^2 W_{i,j} = \min_{(Y^T DY=I)} \operatorname{trace}(Y^T LY),$$

Laplacian Eigenmaps

- Given: $\min_{(Y^T DY=I)} \frac{1}{2}\sum_{i,j} \| y_i - y_j \|^2 W_{i,j} = \min_{(Y^T DY=I)} \mathrm{trace}(Y^T LY),$
- Basic results from linear algebra show that the optimal mapping can be obtained by solving generalized eigenvalue problem **Lx** = λ **Dx**, under the above constraint.
- In particular, coordinates for the mapped vector y_i can be extracted from the ith coordinates of the d eigenvectors with smallest nonzero eigenvalues.

Laplacian Eigenmaps

- Additional Details:
  - Estimation of intrinsic data dimensionality d
  - Selection of local neighborhood resolution parameter k
- Possible Extension:
  - Approximate Nearest Neighbor Selection algorithms for managing larger data sets

Basic Network Construction Workflow

- Starting from from the N x D gene expression data matrix **X**, derive an N x d matrix **Y**, (d < D) using Laplacian Eigenmaps (or another dimensionality reduction technique).
- Construct an N x N matrix **W*** capturing pairwise Euclidean distances between row vectors of **Y** ('reduced gene profiles').
- Apply a statistical significance-based threshold to the elements of **W*** to obtain a network (adjacency matrix) representation.

Network Derivation

- Given:
  - Target p-value α
  - N x N matrix **W*** capturing pairwise Euclidean distances between mapped gene expression profiles in reduced dimensional space.
- Estimate: Distance Threshold
  - Run (scaled) random data through workflow
  - Rank mapped data space pairwise distances
  - Select distance threshold that excludes upper $(1-\alpha)$ fraction of observed values

Basic Network Construction Workflow

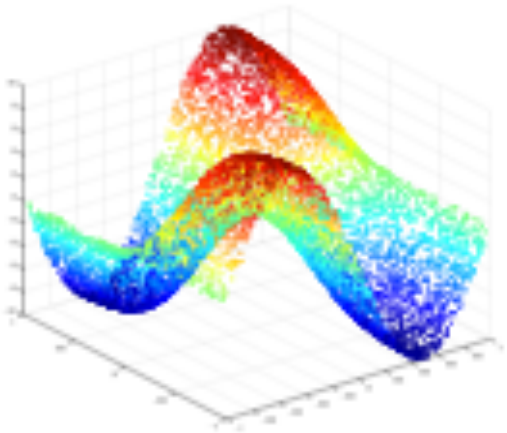
- Starting from from the N x D gene expression data matrix **X**, derive an N x d matrix **Y**, (d < D) using Laplacian Eigenmaps (or another dimensionality reduction technique).
- Construct an N x N matrix **W*** capturing pairwise Euclidean distances between row vectors of **Y** ('reduced gene profiles').
- Apply a statistical significance-based threshold to the elements of **W*** to obtain a network (adjacency matrix) representation.

Implementation Notes

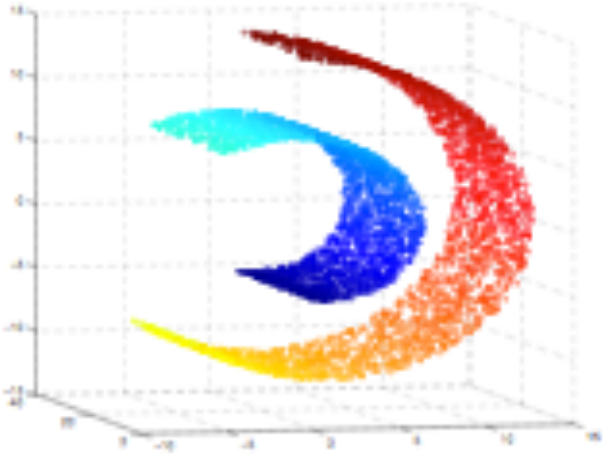
- Overall Aims:
  - Platform-independent, desktop hardware
  - Free and open source
- Implementation Languages:
  - Python (platform libraries, SciPy + matplotlib)
  - C/C++ (core algorithm implementations)
- Libraries:
  - Basic Linear Algebra
  - Network Analysis and Visualization (iGraph)

Validation and Testing

- Laplacian Eigenmaps Implementation
  - Compare to established implementation (MATLAB DR Toolbox) over 4 published synthetic data sets.



(c) Twinpeaks dataset.



(d) Broken Swiss roll dataset.

Validation and Testing

- Integrated Network Reconstruction
  - Compare to published results for leading ARACNE network reconstruction method over:
    - Synthetic Gene Expression Data Set (allows objective scoring of 'true' edge recovery)
    - Well-studied biological data set (compare network reconstruction around MYC oncogene)
  - Compare results Laplacian Eigenmaps-based networks with those derived from original data, PCA-processed data.

Project Schedule and Milestones

- Phase I: Laplacian Eigenmaps + Network Reconstruction
  - Target Date: early December 2010
  - Milestones:
    - Implementation and focused validation of Laplacian Eigenmaps
    - Implementation of statistical significance-based distance matrix thresholding for network construction.

Project Schedule and Milestones

- Phase II: Integrated Testing of Network Reconstruction + Possible Extensions
  - Target Date: end of March 2011
  - Milestones:
    - Integrated testing of network reconstruction
    - Comparison of results obtained using nonlinear dimensionality reduction (LE), linear dimensionality reduction (PCA), and original data
    - Possible: Approximate Nearest Neighbor Algorithm
    - Possible: Diffusion Maps

Deliverables

- Technical report outlining:
  - Problem and general approach
  - Algorithm implementation notes
  - Validation and testing results, including comparative assessment of nonlinear dimensionality reduction in biological network reconstruction
- Source Code – together with data sets and scripts for reproducing results presented in technical report.

References

- M. Belkin and P. Niyogi, *Laplacian Eigenmaps for dimensionality reduction and data representation, Neural Computation.* **15** (2004), No. 6, 1373-1396
- M. Ehler, V. Rajapakse, B. Zeeberg, B. Brooks, J. Brown, W. Czaja, and R. F. Bonner, *Analysis of temporal-spatial co-variation within gene expression microarray data in an organogenesis model*. 6th International Symposium on Bioinformatics Research and Applications (ISBRA'10), Lecture Notes in Bioinformatics, Springer Verlag, 2010
- A. A. Margolin, I Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R Dalla Favera, A. Califano, *ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.* BMC Bioinformatics. 2006 Mar 20;**7** Suppl 1:S7.
- van der Maaten, Postma, van den Herik, *Dimensionality Reduction: A Comparative Review.* Tilburg Centre for Creative Computing Technical Report 2009-005
- Zeeberg B, Qin H, Narasimhan S, et al. *High-Throughput GoMiner, an 'industrial-strength' integrative gene ontology tool for interpretation of multiple-microarray experiments, with application to studies of Common Variable Immune Deficiency (CVID).* BMC Bioinformatics 2005; Jul 5; **6**:168.