<table>
<thead>
<tr>
<th>Project</th>
<th>Psets</th>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
<th>Lec</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe your previous research, areas of interest in computational biology, type of project that best fits your interests. Post in a profile that lets your classmates know you and find potential partners. <strong>Project profile due Tue 9/26</strong>&lt;br&gt;Identify previous project proposals, recent papers, and potential partners that match your areas of interest. List initial project ideas and partners. <strong>Project area/team due Tue 10/3</strong>&lt;br&gt;Form teams of two, specify project goals, division of work, milestones, datasets, challenges. Prepare slide presentation for the class and the mentors. <strong>Project proposal due Tue 10/19. Presented on Fri 10/20</strong></td>
<td>PS1 out on:L1-L5</td>
<td>1</td>
<td>Thu, Sep 7</td>
<td>Introduction</td>
<td>L1</td>
<td>Intro: Biology, Algorithms, Machine Learning, Course Overview</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fri, Sep 8</td>
<td></td>
<td>R1</td>
<td>Recitation 1: Biology and Probability Review</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue, Sep 12</td>
<td></td>
<td>L2</td>
<td>Alignment I: Dynamic Programming, Global and local alignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Thu, Sep 14</td>
<td></td>
<td>L3</td>
<td>Alignment II: Database search, Rapid string matching, BLAST, BLOSUM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fri, Sep 15</td>
<td></td>
<td>R2</td>
<td>Recitation 2: Deriving Parameters of Alignment, Multiple Alignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue, Sep 19</td>
<td></td>
<td>R4</td>
<td>Hidden Markov Models Part 1: Evaluation/Parsing, Viterbi, Forward algorithms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Thu, Sep 21</td>
<td></td>
<td>L5</td>
<td>Hidden Markov Models Part 2: Posterior Decoding, Learning, Baum-Welch</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fri, Sep 22</td>
<td></td>
<td></td>
<td>No classes - student holiday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mon, Sep 25</td>
<td></td>
<td></td>
<td></td>
<td><strong>Project Intro:</strong> about the projects, self introductions, mentor intro, example projects, teamwork 32D-507</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Module II: Gene Expression and Epigenomics</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td>Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L6</td>
<td>Expression Analysis: Clustering/Classification, K-means, Hierarchical, Bayesian</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L7</td>
<td>Transcript structure: GenScan, RNA-seq, Mapping, De novo Assembly, Diff Expr</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>R3</td>
<td>Recitation 3: Affinity Propagation Clustering and Random Forest Classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L8</td>
<td>Epigenomics: ChIP-Seq, Read mapping, Peak calling, IDR, Chromatin states</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L9</td>
<td>Three-dimensional chromatin interactions: 3C, 5C, HiC, CHIA-Pet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>R4</td>
<td>Recitation 4: ENCODE, Epigenome Roadmap, ChromHMM, ChromImpute</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Project Planning:</strong> research areas, initial ideas, type of project, mentor matching, finding partners 32D-507</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Module III: Regulatory Genomics and Networks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td>Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L10</td>
<td>Regulatory Motifs: Discovery, Representation, PBMs, Gibbs Sampling, EM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L11</td>
<td>Recitation 5: Gapped Motif Discovery, DNashape, PBMs, Selex</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L12</td>
<td>Network structure, centrality, SVD, sparse PCA, L1/L2, modules, diffusion kernels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L13</td>
<td>Deep Learning, Neural Nets, Convolutional NNs, Recursive NNs, Autoencoder</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L14</td>
<td>Recitation 6: Networks review, Recommendation systems, EHR, PhEwas</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Project feedback:</strong> Prepare 2-3 slide presentation of your term project for your mentor. 32D-507 at 4-5pm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Module IV: Population Genetics and Disease Genomics</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td>Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L15</td>
<td>Population genetics: Linkage disequilibrium, pop struct, 1000genomes, allele freq</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L16</td>
<td>Disease Association Mapping, GWAS, organismal phenotypes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L17</td>
<td>Quantitative trait mapping, molecular traits, eQTLs, mediation analysis, IMWAS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L18</td>
<td>Linkage Disequilibrium, Haplotype Phasing, Genotype Imputation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L19</td>
<td>Missing Heritability, Complex Traits, Interpret GWAS, Rank-based enrichment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L20</td>
<td>Recitation 7: Rare Variants, ExAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L21</td>
<td>Comparative genomics and evolutionary signatures</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L22</td>
<td>Genome Scale Evolution, Genome Duplication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Module V: Comparative Genomics and Evolution</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td>Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L23</td>
<td>Phylogenetics: Molecular evolution, Tree building, Phylogenetic inference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L24</td>
<td>Phylogenomics: Gene/species trees, reconciliation, coalescent, ARGs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L25</td>
<td>Recitation 9: Phylogenetic distance metrics, Coalescent Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td><strong>Module VI: Current Research Directions</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td></td>
<td>Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L26</td>
<td>Single-cell genomics: technology, analysis, microfluidics, applications, insights</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L27</td>
<td>Mining human phenotypes, Phewan, UK Biobank, meta-phenotypes+imputation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>R10</td>
<td>Recitation 10: Project Feedback, results, interpretation, directions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L28</td>
<td>Cancer Genomics, Single-cell Sequencing, Tumor-Immune Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L29</td>
<td>Genome Engineering with CRISPR/Cas9 and related technologies</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>R11</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L30</td>
<td>Final Presentations - Part I (11am), 32-G8 reading room</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tue 10/10</td>
<td></td>
<td>L31</td>
<td>Final Presentations - Part I (1pm), 32-141 reading room</td>
</tr>
</tbody>
</table>

* readings refer to chapters in compiled 2016 scribe notes, available in the materials folder on Stellar

** recitation topics will be adjusted to respond to lecture and student needs
Module III: Regulation, Epigenomics, Networks

**Computational Foundations:** Machine learning, dealing with noisy data
- L6: Clustering/classification: Supervised/unsupervised learning
- L10: Expectation Maximization (EM), Gibbs Sampling, Information theory
  - L11: Network algorithms, probabilistic interpretation, linear algebra
- L12: Deep Learning, Neural Nets, CNNs, RNNs

**Biological frontiers:** Gene Regulation, Regulatory Systems Genomics
- L6: Gene expression analysis.
- L8,9: Epigenomics and Chromatin state.
- L10: Regulatory motif discovery.
- L11: Biological Network inference and analysis.

---

<table>
<thead>
<tr>
<th>Module III: Regulatory Genomics and Networks</th>
<th>No Classes - Columbus Day Holiday</th>
</tr>
</thead>
<tbody>
<tr>
<td>L10</td>
<td>Regulatory Motifs: Discovery, Representation, PBM, Gibbs Sampling, EM</td>
</tr>
<tr>
<td>R5</td>
<td>Recitation 5: Gapped Motif Discovery, DNAs, PBM, Selex</td>
</tr>
<tr>
<td>L11</td>
<td>Network structure, centrality, SVD, sparse PCA, L1/L2, modules, diffusion kernels</td>
</tr>
<tr>
<td>L12</td>
<td>Deep Learning, Neural Nets, Convolutional NNs, Recurrent NNs, Autoencoder</td>
</tr>
<tr>
<td>R6</td>
<td>Recitation 6: Networks review, Recommendation systems, EHR, PheWAS</td>
</tr>
</tbody>
</table>

Project feedback: Prepare 2-3 slide presentation of your term project for your mentor. 32D-507 at 4-5pm
Goals for today: Network analysis

1. **Introduction to networks**
   - Network types: regulatory, metab., signal., interact., func.
   - Bayesian (probabilistic) and Algebraic views

2. **Network Centrality Measures**
   - Local centrality metrics (degree, betweenness, closeness, etc)
   - Global centrality metrics (eigenvector centrality, page-rank)

3. **Linear Algebra Review**
   - Eigenvector and singular vector decomposition
   - Low rank approximations, Wigner semicircle law

4. **Sparse Principal Component Analysis**
   - Lasso and Elastic lasso
   - PCA and Sparse PCA

5. **Network Communities and Modules**
   - Guilt by association
   - Maximum cliques, density-based modules and spectral clustering

6. **Network Diffusion Kernels and Deconvolution**
   - Network diffusion kernels
   - Network deconvolution
The multi-layered organization of information in living systems

- **transcriptome**: Genes
  - DNA
  - RNA
  - Proteins
  - ncRNA
  - mRNA
  - miRNA
  - piRNA

- **genome**: Genes
  - DNA
  - cis-regulatory elements

- **epigenome**: Genes
  - CHROMATIN
  - histones

- **proteome**: Proteins
  - transcription factors
  - signaling proteins
  - metabolic enzymes

**Diagram**

- DNA
- RNA
- Proteins
- Genes
- CHROMATIN
- histones
- cis-regulatory elements
- piRNA
- Metabolic Enzymes
- Transcription factors
- Signaling proteins
- ncRNA
- mRNA
- miRNA
Biological networks at all cellular levels

Dynamics

RNA

Proteins

Translation

Modification

Transcription

Transcriptional gene regulation

Post-transcriptional gene regulation

Protein & signaling networks

Metabolic networks

Genome

Dynamics

RNA

Proteins

Translation

Modification

Transcription

Transcriptional gene regulation

Post-transcriptional gene regulation

Protein & signaling networks

Metabolic networks
Five major types of biological networks

**Regulatory network**
- Transcription factors (TF)
- Regulatory network

**Metabolic network**
- Enzymes
- Metabolites

**Signaling network**
- Receptors
- Protein complex

**PPI, Protein interaction network**
- Protein complex

**Functional network (Co-expression)**
- Undirected, weighted
Information exchange across networks

- Signaling network
  - Receptors
  - Signaling protein
  - Transcription factors (TF)
  - Activate TFs
  - Form TF complexes

- Regulatory network
  - Transcribe enzymes
  - Transcribe proteins

- Metabolic network
  - Protein interaction network
Network applications and challenges

1. Element Identification (motif finding lecture)

- Regulators
- Regulatory Motifs
- Target genes

2. Using networks to predict cellular activity

- Predict expression levels
- Predict gene ontology (GO) functional annotation terms

3. Inferring networks from functional data

- Activity patterns

4. Network Structure Analysis

- Hubs (degree-distribution)
- Network motifs
- Functional modules
Probabilistic networks and graphical model

• There are several types of networks, with different meanings, and different applications

• Networks as graphical models:
  – modeling joint probability distribution of variables using graphs
  – Bayesian networks (directed), Markov Random Fields (undirected)

\[
X_{S_1} \perp X_{S_3} | X_{S_2}
\]
Physical and Relevance Networks

• Physical Networks:
  – edges represent “physical interaction” among nodes
  – Example: physical regulatory networks

• Relevance Networks:
  – edge weights represent node similarities
  – Example: functional regulatory networks
**Representing Networks as Graphs**

- **Weighted graphs**: weights associated to every edge, generally positive
  
  ![Weighted graph example](image)

- **Multigraphs (Pseudographs)**: multiple edges can exist among nodes
  
  ![Multigraph example](image)

- **Digraphs**: edges have directions
  
  ![Digraph example](image)

- **Simple graphs**: no multiple edges or self-loops
  
  ![Simple graph example](image)
Matrix representation of networks

• A matrix representation of a network:
  – **Unweighted network**: binary adjacency matrix
  – **Weighted network**: real-valued matrix

A | B | C | D | E
---|---|---|---|---
A | 0 | 1.5 | 0 | 0 | 0
B | 1.5 | 0 | 3.4 | 0 | 0
C | 0 | 3.4 | 0 | 2.1 | 0.5
D | 0 | 0 | 2.1 | 0 | 0.7
E | 0 | 2.1 | 0.5 | 0.7 | 0

| Degree | 1.5 | 4.9 | 6 | 2.8 | 3.3 |
Goals for today: Network analysis

1. Introduction to networks
   – Network types: regulatory, metab., signal., interact., func.
   – Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   – Local centrality metrics (degree, betweenness, closeness, etc)
   – Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   – Eigenvector and singular vector decomposition
   – Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   – Lasso and Elastic lasso
   – PCA and Sparse PCA

5. Network Communities and Modules
   – Guilt by association
   – Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   – Network diffusion kernels
   – Network deconvolution
Centrality Measures in Networks

**Question:** how important is a node/edge to the structural characteristics of the system?

- **High connectivity node**
- **Low degree but still important?**
Degree Centrality

- Example:

- Degree Centrality can be similarly defined for
  - Directed graphs, in- and out- degrees
  - Weighted graphs, weighted degrees
Betweenness centrality

- The number of shortest paths in the graph that pass through the node divided by the total number of shortest paths.

\[ BC(k) = \sum_i \sum_j \frac{\rho(i,k,j)}{\rho(i,j)}, \quad i \neq j \neq k \]

- Nodes with a high betweenness centrality control information flow in a network.

- Edge betweenness is defined similarly.

**Shortest paths are:**
- AB, AC, ABD, ABE, BC, BD, BE, CBD, CBE, DBE
- B has a BC of 5
Closeness Centrality

- The normalised inverse of the sum of topological distances in the graph.

$$CC(i) = \frac{N - 1}{\sum_j d(i, j)}$$

- Node B is the most central one in spreading information from it to the other nodes in the network.

- DC, BC and CC all agree

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$\sum_{j=1}^n d(i, j)$</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6</td>
<td>0.67</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>1.00</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>0.67</td>
</tr>
<tr>
<td>D</td>
<td>7</td>
<td>0.57</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>0.57</td>
</tr>
</tbody>
</table>
When closeness centrality and degree centrality are different

- A is the most central according to the degree
- B is the most central according to closeness and betweenness

Which is the most central node?
Eigenvector Centrality: Extending the Concept of Degree

• Make $x_i$ proportional to the average of the centralities of its $i$’s network neighbors

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j$$

where $\lambda$ is a constant. In matrix-vector notation we can write

$$x = \frac{1}{\lambda} Ax$$

In order to make the centralities non-negative we select the eigenvector corresponding to the principal eigenvalue (Perron-Frobenius theorem).
Goals for today: Network analysis

1. Introduction to networks
   – Network types: regulatory, metab., signal., interact., func.
   – Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   – Local centrality metrics (degree, betweenness, closeness, etc)
   – Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   – Eigenvector and singular vector decomposition
   – Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   – Lasso and Elastic lasso
   – PCA and Sparse PCA

5. Network Communities and Modules
   – Guilt by association
   – Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   – Network diffusion kernels
   – Network deconvolution
Linear Algebra Review: Eigenvalues & Eigenvectors

• **Eigenvectors** (for a square $m \times m$ matrix $S$)

  - \[ \mathbf{Sv} = \lambda \mathbf{v} \]

  (right) eigenvector \hspace{1cm} \text{eigenvalue}
  \[ \mathbf{v} \in \mathbb{R}^m \neq \mathbf{0} \hspace{1cm} \lambda \in \mathbb{R} \]

• **How many eigenvalues** are there at most?

  \[ \mathbf{Sv} = \lambda \mathbf{v} \iff (\mathbf{S} - \lambda \mathbf{I}) \mathbf{v} = \mathbf{0} \]

  only has a non-zero solution if \[ |\mathbf{S} - \lambda \mathbf{I}| = 0 \]

  this is a $m$-th order equation in $\lambda$ which can have **at most** $m$ **distinct solutions** (roots of the characteristic polynomial) - can be complex even though $S$ is real.

**Example**

\[
\begin{pmatrix}
6 & -2 \\
4 & 0
\end{pmatrix}
\begin{pmatrix}
1 \\
2
\end{pmatrix} =
\begin{pmatrix}
2 \\
4
\end{pmatrix}
= 2 \begin{pmatrix}
1 \\
2
\end{pmatrix}
\]
Matrix-vector multiplication

- Example:

\[
S = \begin{bmatrix}
3 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

has eigenvalues 3, 2, 0 with corresponding eigenvectors:

\[
v_1 = \begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix}, \quad v_2 = \begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}, \quad v_3 = \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix}
\]

On each eigenvector, \( S \) acts as a multiple of the identity matrix: but as a different multiple on each.

Any vector (say \( x = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \)) can be viewed as a combination of the eigenvectors:

\[
x = 2v_1 + 4v_2 + 6v_3
\]
Matrix vector multiplication

• Thus a matrix-vector multiplication such as $Sx$ ($S$, $x$ as in the previous slide) can be rewritten in terms of the eigenvalues/vectors:

$$Sx = S(2v_1 + 4v_2 + 6v_3)$$
$$Sx = 2Sv_1 + 4Sv_2 + 6Sv_3 = 2\lambda_1 v_1 + 4\lambda_2 v_2 + 6\lambda_3 v_3$$

• Even though $x$ is an arbitrary vector, the action of $S$ on $x$ is determined by the eigenvalues/vectors.

• Observation: the effect of “small” eigenvalues is small.
Eigenvalues & Eigenvectors

- For symmetric matrices, eigenvectors for distinct eigenvalues are **orthogonal**
  \[ S v_{\{1,2\}} = \lambda_{\{1,2\}} v_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \implies v_1 \cdot v_2 = 0 \]

- All eigenvalues of a real symmetric matrix are **real**.
  
  for complex \( \lambda \), if \( |S - \lambda I| = 0 \) and \( S = S^T \) \( \implies \lambda \in \mathbb{R} \)

- All eigenvalues of a positive semidefinite matrix are **non-negative**
  
  \( \forall w \in \mathbb{R}^n, w^T Sw \geq 0 \), then if \( Sv = \lambda v \) \( \implies \lambda \geq 0 \)
Example- Eigenvalues & Eigenvectors

• Let

\[ S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \]

Real, symmetric.

• Then

\[ S - \lambda I = \begin{bmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{bmatrix} \Rightarrow (2 - \lambda)^2 - 1 = 0. \]

• The eigenvalues are 1 and 3 (nonnegative, real).

• The eigenvectors are orthogonal (and real):

\[ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

Plug in these values and solve for eigenvectors.
Eigen/diagonal Decomposition

• Let $S \in \mathbb{R}^{m \times m}$ be a **square** matrix with $m$ linearly independent eigenvectors (a “non-defective” matrix)

\[
S = U \Lambda U^{-1}
\]

• **Theorem**: Exists an eigen decomposition

\[
S = U \Lambda U^{-1}
\]

– (cf. matrix diagonalization theorem)

• Columns of $U$ are **eigenvectors** of $S$

• Diagonal elements of $\Lambda$ are **eigenvalues** of $S$

\[
\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}
\]
Why is diagonal decomposition possible

- Let \( U \) have the eigenvectors as columns: 
  \[
  U = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}
  \]

Then, \( SU \) can be written

\[
SU = S \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} = \begin{bmatrix} \lambda_1 v_1 & \ldots & \lambda_n v_n \end{bmatrix} = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} \begin{bmatrix} \lambda_1 & \ldots & \lambda_n \end{bmatrix}
\]

Thus \( SU = U \Lambda \), and \( S(UU^{-1}) = U \Lambda U^{-1} \)

And \( S = U \Lambda U^{-1} \).
Diagonal decomposition - example

- Recall \( S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \); \( \lambda_1 = 1, \lambda_2 = 3 \).

The eigenvectors \( \begin{pmatrix} 1 \\ -1 \end{pmatrix} \) and \( \begin{pmatrix} 1 \\ 1 \end{pmatrix} \) form \( U = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \).

Inverting, we have \( U^{-1} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix} \).

Then, \( S = U \Lambda U^{-1} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix} \).
Example continued

- Let’s divide \( U \) (and multiply \( U^{-1} \)) by \( \sqrt{2} \)

Then, \( S = \)

\[
\begin{bmatrix}
1/\sqrt{2} & 1/\sqrt{2} \\
-1/\sqrt{2} & 1/\sqrt{2}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & 3
\end{bmatrix}
\begin{bmatrix}
1/\sqrt{2} & -1/\sqrt{2} \\
1/\sqrt{2} & 1/\sqrt{2}
\end{bmatrix}
\]

\( Q \quad \Lambda \quad (Q^{-1}=Q^T) \)

\( Q^{-1}=Q^T \rightarrow Q^TQ=I \rightarrow \text{Orthogonal matrix} \)

Every column has unit length and is perpendicular to all others.
Symmetric Eigen Decomposition

• If $S \in \mathbb{R}^{m \times m}$ is a symmetric matrix:

• **Theorem:** Exists a (unique) eigen decomposition
  
  $S = Q \Lambda Q^T$

• where $Q$ is orthogonal:
  
  – $Q^{-1} = Q^T$
  
  – Columns of $Q$ are normalized eigenvectors
  
  – Columns are orthogonal.
  
  – (everything is real)
Singular Value Decomposition

For an $m \times n$ matrix $A$ of rank $r$ there exists a factorization (Singular Value Decomposition = SVD) as follows:

$$A = U \Sigma V^T$$

The columns of $U$ are orthogonal eigenvectors of $A A^T$.
The columns of $V$ are orthogonal eigenvectors of $A^T A$.
Eigenvalues $\lambda_1 \ldots \lambda_r$ of $A A^T$ are the eigenvalues of $A^T A$.

$$\sigma_i = \sqrt{\lambda_i}$$

$$\Sigma = \text{diag}(\sigma_1 \ldots \sigma_r)$$

Singular values.
Singular Value Decomposition

- Illustration of SVD dimensions and sparseness
Singular Value Decomposition-example

- Let \( A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \)

Thus \( m=3, \ n=2 \). Its SVD is

\[
\begin{bmatrix}
0 & 2/\sqrt{6} & 1/\sqrt{3} \\
1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\
1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & \sqrt{3} \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
1/\sqrt{2} & 1/\sqrt{2} \\
1/\sqrt{2} & -1/\sqrt{2}
\end{bmatrix}
\]

Typically, the singular values arranged in decreasing order.
Low-rank Approximation

• SVD can be used to compute optimal **low-rank approximations**.

• Approximation problem: Find $A_k$ of rank $k$ such that

$$A_k = \min_{X: \text{rank}(X) = k} \| A - X \|_F$$

**Frobenius norm** (aka Euclidian norm)

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}.$$ 

$A_k$ and $X$ are both $m \times n$ matrices.

Typically, want $k << r$. 
Low-rank Approximation

• Solution via SVD

\[ A_k = U \text{ diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0) V^T \]

set smallest \( r-k \) singular values to zero

\[ A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \]

\text{column notation: sum of rank 1 matrices}

• Error:

\[ \min_{X: \text{rank}(X) = k} \| A - X \|_F = \| A - A_k \|_F = \sigma_{k+1} \]
How to choose rank? Look for non-random eigenvalues/eigenvectors

- **Question**: does the network (matrix) look like a random matrix? Does it some non-random structure?

- **Wigner semicircle law (GUE)**: distribution of singular values of a random matrix (GUE) form a semi-circle

- **Tracy–Widom distribution**: the probability distribution of the largest eigenvalue
Looking for non-random eigenvalues

- **Example:** Distribution of eigenvalues of a ‘non-random’ matrix:

Define p-value for each PC
Consider significant PCs in the analysis
Structural Inference using SVD-example

Matrix: $U_1$ and $V_1$

Singular values: Meaningful combinations of genes x expts

Eigen-experiments: $U_1$ and $V_1$

$n$ genes: Singular values

$m$ experiments: Meaningsful combinations of genes x expts
Structural Inference using SVD-example

- **Matrix**
- **Singular values**
- **$U_1$**
- **$V_1$**
Structural Inference using SVD-example

Matrix

Singular values

$U_2$

$V_2$
Goals for today: Network analysis

1. Introduction to networks
   – Network types: regulatory, metab., signal., interact., func.
   – Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   – Local centrality metrics (degree, betweenness, closeness, etc)
   – Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   – Eigenvector and singular vector decomposition
   – Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   – Lasso and Elastic lasso
   – PCA and Sparse PCA

5. Network Communities and Modules
   – Guilt by association
   – Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   – Network diffusion kernels
   – Network deconvolution
Sparse Principal Component Analysis

Gene Expression Data
(RNA-Seq, Microarray, ...)

- $n = 20k$ genes, $m = 100$ arrays
- $n \gg m$
PCA on Expression Data

\[ U \times D \times V^T = E \]

- Each eigen-gene is expressed only in the corresponding eigen-array with the corresponding eigen-expression level.

(Alter et al., 2000)
Drawbacks of PCA

- PCs (eigen-experiments) are linear combinations of all \( n \) variables (genes).
  - Each PC corresponds to a loading vector (columns of \( V \))
  - Loadings = coefficients corresponding to variables in the linear combination
- Difficult to interpret
Sparse PCA

• Idea: Reduce the number of explicitly used variables (genes).

• Approach: Modify PCA so that PCs have sparse loadings = sparse PCA (SPCA)

• Writes PCA as a regression-type optimization problem.

• Uses lasso (Least Absolute Shrinkage and Selection Operator)
  – Tibshirani 1996
  – Both variable selection and regularization
  – Produces sparse models

• Result: Modified PCs with sparse loadings.
Linear Regression Problem

- Input variables \( x = (1, x_1, \ldots, x_p) \)
- Response variable \( y \equiv f(x) + \varepsilon \)
- Regression coefficients \( \beta = (\beta_0, \beta_1, \ldots, \beta_p)^T \)
- Multivariate linear model

\[
f(x_1, x_2, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = x \beta
\]
Linear Regression Problem

- N observations, p predictors.
- Goal: Estimate the coefficients, $\beta$. 

\[
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N \\
\end{bmatrix} = 
\begin{bmatrix}
1 & x_{11} & x_{12} & \cdots & x_{1p} \\
1 & x_{21} & x_{22} & \cdots & x_{2p} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{N1} & x_{N2} & \cdots & x_{Np} \\
\end{bmatrix}
\begin{bmatrix}
\beta_0 \\
\beta_1 \\
\vdots \\
\beta_p \\
\end{bmatrix} + 
\begin{bmatrix}
\epsilon_1 \\
\epsilon_2 \\
\vdots \\
\epsilon_N \\
\end{bmatrix}
\]

Training Data Coefficients Error
Least Squares Solution

Training Set \( D = \{(X_i, y_i)\}_{i=1}^{n} \)

Residual Sum of Squares \( RSS(\beta) \equiv \sum_{i=1}^{n} (y_i - \beta X_i)^2 \)

Objective: Find \( \hat{\beta} = \arg \min_{\beta} \{ RSS(\beta) \mid D \} \)
Sparse PCA solution with Lasso penalty

Training Set \[ D = \{(X_i, y_i)\}_{i=1}^n \]

Residual Sum of Squares \[ \text{RSS}(\beta) = \sum_{i=1}^n (y_i - \beta X_i)^2 \]

Lasso penalty \[ L_1(\beta) = \lambda \sum_{j=1}^p |\beta_j|, \quad \lambda \geq 0 \]

Objective: Find \[ \hat{\beta}_{\text{lasso}} = \arg \min_{\beta} \{ \text{RSS}(\beta) + L_1(\beta) \mid D \} \]

• Pros:
  – Lasso continuously shrinks coefficients toward zero.
  – Produces a sparse model.
  – Variable selection method.

• Cons:
  – #selected variables limited by \( n \), number of observations.
    e.g. microarray expression data \( n(\text{arrays}) \ll p(\text{genes}) \)
  – Selects only one of the highly correlated variables, does not care which one is in the final model.
L1 (Lasso) vs. L2 (Ridge) regularization

- L1 regularization (lasso)
  \[
  w^* = \arg \min_w \sum_j \left( t(x_j) - \sum_i w_i h_i(x_j) \right)^2 + \lambda \sum_{i=1}^k |w_i|
  \]

- L2 regularization (ridge)
  \[
  w^* = \arg \min_w \sum_j \left( t(x_j) - \sum_i w_i h_i(x_j) \right)^2 + \lambda \sum_{i=1}^k w_i^2
  \]

- L1 pros:
  - More robust (outliers don’t matter more than small diffs)
- L1 cons:
  - Less stable gradient ascent
  - Multiple solutions

- L2 pros:
  - Always one solution
  - More stable gradient ascent
- L2 cons:
  - Less robust (square diffs, outliers have strong effect)
Elastic Net Solution

Lasso penalty  \( L_1(\beta) = \lambda_1 \sum_{j=1}^{p} |\beta_j|, \quad \lambda_1 \geq 0 \)

Ridge penalty  \( L_2(\beta) = \lambda_2 \sum_{j=1}^{p} |\beta_j|^2, \quad \lambda_2 \geq 0 \)

Objective: Find \( \hat{\beta}_{ridge} = \arg \min_{\beta} \{RSS(\beta) + L_2(\beta) + L_1(\beta) | D\} \)

• Pros:
  – Combines L1 (lasso) and L2 (ridge) regression
  – Limitation of lasso removed by ridge constraint. All variables are included in the model.
  – Grouping effect:
    • Selects a group of highly correlated variables once one variable among them is selected.
    (lasso selects only one of them, does not care which.)
SPCA

• Goal:
  – Construct a regression framework in which PCA can be reconstructed exactly.
  – Use lasso/ridge/elastic-net to construct modified PCs with sparse loadings.
Reconstruction of PCA in a Regression Framework

Idea: Each PC is a linear combination of the p variables. Its loadings can be recovered by regression PC on the p variables.
Theorem- solving PCA by regression

Let \( X = UDV^T \).

\( \forall i \), denote \( Y_i = U_i D_{i,i} \). \( Y_i \) is the \( i \)th principal component of \( X \).

\( \forall \lambda > 0 \), suppose \( \hat{\beta}_{\text{ridge}} \) is the ridge estimate given by

\[
\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} |Y_i - X\beta|^2 + \lambda |\beta|^2 .
\]

Let \( \hat{v} = \frac{\hat{\beta}_{\text{ridge}}}{|\hat{\beta}_{\text{ridge}}|} \), then \( \hat{v} = V_i \).
Reconstruction of PCA in a Regression Framework

• By theorem 1, we can reconstruct the loadings of PCs exactly by a linear regression problem.
  – not an alternative to PCA as it uses its results.

• Ridge penalty does not penalize the coefficients, but ensure the reconstruction of PCs.

• Now, add lasso penalty to the problem to penalize for the absolute values of coefficients.
Construction of SPCA in a Regression Framework

Let $X = UDV^T$.

$\forall i$, denote $Y_i = U_i D_{i,i} \cdot Y_i$ is the $i^{th}$ principal component of $X$.

Solve $\hat{\beta} = \arg \min_{\beta} |Y_i - X\beta|^2 + \lambda |\beta|^2 + \lambda_1 |\beta|$.

$\hat{V}_i = \frac{\hat{\beta}}{|\hat{\beta}|} \approx V_i$

$X\hat{V}_i \approx Y_i = i^{th}$ principal component

$\hat{V}_i = \text{sparse loading, } X\hat{V}_i = i^{th}$ sparse principal component
Simulation Example:
PCA vs. SPCA

• Data points $X = (X_1, X_2, ..., X_{10})$
  – 10 variables

• Model to generate data:
  – 3 hidden factors: $V_1, V_2, V_3$
  – $V_1 \sim N(0, 290)$
  – $V_2 \sim N(0, 300)$
  – $V_3 = -0.3\ V_1 + 0.925\ V_2 + e,\ e \sim N(0, 1)$
  \[
  \begin{align*}
  &- X_i = V_1 + e_i^1,\ e_i^1 \sim N(0, 1), \ i=1,2,3,4 \\
  &- X_i = V_2 + e_i^2,\ e_i^2 \sim N(0, 1), \ i=5,6,7,8 \\
  &- X_i = V_3 + e_i^3,\ e_i^3 \sim N(0, 1), \ i=9,10 \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &\quad \text{4 vars associated with } V_1 \\
  &\quad \text{4 vars associated with } V_2 \\
  &\quad \text{2 vars associated with } V_3 \\
  \end{align*}
  \]
Simulation Example: PCA vs. SPCA

- How many observations?
  - PCA and SPCA performed on exact covariance matrix. => infinitely many data points.

  - We expect to derive 2 PCs with right sparse loadings:
    
    - One from \((X_5, X_6, X_7, X_8)\) recovering \(V_2\)
    - One from \((X_1, X_2, X_3, X_4)\) recovering \(V_1\)
# Table of Loadings

<table>
<thead>
<tr>
<th></th>
<th>PCA PC1</th>
<th>PCA PC2</th>
<th>PCA PC3</th>
<th>SPCA PC1</th>
<th>SPCA PC2 (λ = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0.116</td>
<td>-0.478</td>
<td>-0.087</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.116</td>
<td>-0.478</td>
<td>-0.087</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.116</td>
<td>-0.478</td>
<td>-0.087</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_4$</td>
<td>0.116</td>
<td>-0.478</td>
<td>-0.087</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_5$</td>
<td>-0.395</td>
<td>-0.145</td>
<td>0.270</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>$X_6$</td>
<td>-0.395</td>
<td>-0.145</td>
<td>0.270</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>$X_7$</td>
<td>-0.395</td>
<td>-0.145</td>
<td>0.270</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>$X_8$</td>
<td>-0.395</td>
<td>-0.145</td>
<td>0.270</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>$X_9$</td>
<td>-0.401</td>
<td>0.010</td>
<td>-0.582</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>-0.401</td>
<td>0.010</td>
<td>-0.582</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Adjusted Variance (%)</td>
<td>60.0</td>
<td>39.6</td>
<td>0.08</td>
<td>40.9</td>
<td>39.5</td>
</tr>
</tbody>
</table>
Goals for today: Network analysis

1. Introduction to networks
   - Network types: regulatory, metab., signal., interact., func.
   - Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   - Local centrality metrics (degree, betweenness, closeness, etc)
   - Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   - Eigenvector and singular vector decomposition
   - Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   - Lasso and Elastic lasso
   - PCA and Sparse PCA

5. Network Communities and Modules
   - Guilt by association
   - Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   - Network diffusion kernels
   - Network deconvolution
Predicting functions of un-annotated genes

• Goal: Predict function of unannotated genes based on “guilt by association”

• Different types of “association”

However most approaches work with “functional networks”

Deng et al 03, Sharan et al 07
Iterative classification algorithm

- Start with an initial assignment of labels
- Repeat iteratively
  - Update relational attributes
  - Re-infer the labels

Neville 03, Getoor 05
Modularity of regulatory networks

- Modular: Graph with densely connected subgraphs
- Genes in modules involved in similar functions and co-regulated
- Modules can be identified using graph partitioning algorithms
  - Markov Clustering Algorithm
  - Girvan-Newman Algorithm
  - Spectral partitioning

Newman PNAS 2007
Definition of a community can be subjective. (unsupervised learning)
Taxonomy of Community Criteria

• Criteria vary depending on the tasks
• Roughly, community detection methods can be divided into 4 categories (not exclusive):
  • **Node-Centric Community**
    – Each node in a group satisfies certain properties
  • **Group-Centric Community**
    – Consider the connections within a group as a whole. The group has to satisfy certain properties without zooming into node-level
  • **Network-Centric Community**
    – Partition the whole network into several disjoint sets
  • **Hierarchy-Centric Community**
    – Construct a hierarchical structure of communities
Complete Mutuality: Clique

- **Clique**: a maximum complete subgraph in which all nodes are adjacent to each other

Nodes 5, 6, 7 and 8 form a clique

- NP-hard to find the maximum clique in a network
- Straightforward implementation to find cliques is very expensive in time complexity
Finding the Maximum Clique

• In a clique of size $k$, each node maintains degree $\geq k-1$
  – Nodes with degree $< k-1$ will not be included in the maximum clique

• Recursively apply the following pruning procedure
  – Sample a sub-network from the given network, and find a clique in the sub-network, say, by a greedy approach
  – Suppose the clique above is size $k$, in order to find out a larger clique, all nodes with degree $\leq k-1$ should be removed.

• Repeat until the network is small enough

• Many nodes will be pruned as social media networks follow a power law distribution for node degrees
Maximum Clique Example

• Suppose we sample a sub-network with nodes \{1-9\} and find a clique \{1, 2, 3\} of size 3.

• In order to find a clique >3, remove all nodes with degree \leq 3 - 1 = 2
  – Remove nodes 2 and 9
  – Remove nodes 1 and 3
  – Remove node 4
Group-Centric Community Detection: Density-Based Groups

• The group-centric criterion requires the whole group to satisfy a certain condition
  – E.g., the group density $\geq$ a given threshold

• A subgraph $G_s(V_s, E_s)$ is a $\gamma$-dense quasi-clique if

\[
\frac{2|E_s|}{|V_s|(|V_s|-1)} \geq \gamma
\]

where the denominator is the maximum number of degrees.

• A similar strategy to that of cliques can be used
  – Sample a subgraph, and find a maximal $\gamma$-dense quasi-clique (say, of size $|V_s|$)
  – Remove nodes with degree less than the average degree

\[
< |V_s| \gamma \leq \frac{2|E_s|}{|V_s|-1}
\]
Network-Centric Community Detection

• Network-centric criterion needs to consider the connections within a network **globally**

• Goal: **partition nodes of a network into disjoint sets**

• Approaches:
  – (1) Clustering based on vertex similarity
  – (2) Latent space models (multi-dimensional scaling)
  – (3) Block model approximation
  – **(4) Spectral clustering**
  – (5) Modularity maximization
Laplacian matrix of a graph G

• $L=D-A$ (Degree matrix minus Adjacency matrix)

• Adjacency Matrix

$$A= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

• Laplacian Matrix

$$L= \begin{bmatrix} K_1 & -A_{ij} \\ -A_{ij} & K_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$
Network modularization-example

A =

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

L =

<table>
<thead>
<tr>
<th>3</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Adjacency Matrix  Laplacian Matrix
Eigen decomposition-example

\[ L = U \Sigma U^{-1} \]

\[ U = \begin{bmatrix}
0.3536 & -0.3825 & 0.2714 & -0.1628 & -0.7783 & 0.0495 & -0.0064 & -0.1426 \\
0.3536 & -0.3825 & 0.5580 & -0.1628 & 0.6066 & 0.0495 & -0.0064 & -0.1426 \\
0.3536 & -0.3825 & -0.4495 & 0.6251 & 0.0930 & 0.0495 & -0.3231 & -0.1426 \\
0.3536 & -0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & 0.6626 \\
0.3536 & 0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & -0.6626 \\
0.3536 & 0.3825 & 0.3514 & 0.5572 & -0.0727 & -0.3466 & 0.3860 & 0.1426 \\
0.3536 & 0.3825 & 0.0284 & -0.2577 & -0.0059 & -0.3466 & -0.7218 & 0.1426 \\
0.3536 & 0.3825 & 0.0000 & 0.0000 & 0.0000 & 0.8416 & -0.0000 & 0.1426
\end{bmatrix} \]

\[ \Sigma = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.3542 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4.0000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4.0000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4.0000 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.6458
\end{bmatrix} \]

- First eigenvalue of Laplacian matrix is always zero.
- Second eigenvector of Laplacian matrix characterizes a network partition.
Laplacian matrix characterizes # of edges between groups

node $i$ in group 1 $\Rightarrow s_i=1$
node $i$ in group 2 $\Rightarrow s_i=-1$

$i$-th component of $L S =$
$= \text{degree } (i) - (\# \text{ edges to group 1}) + (\# \text{ edges to group 2})$

Diagonal term of Laplacian
Off-diagonal terms of Laplacian

$= 4 - 3 + 1 = 2 = 2 * (\# \text{ edges going out of group 1})$
Laplacian matrix characterizes # of edges between groups

node $i$ in group 1 $\Rightarrow s_i = 1$
node $i$ in group 2 $\Rightarrow s_i = -1$

# of edges between groups $= \frac{1}{4} s^t L s$

- Choose vector $s$ to minimize the error term
- A trivial solution: if $s = (1, 1, \ldots, 1)$, error is zero.
- A non-trivial solution: $s$ parallel to the second eigenvector of $L$ (why?)
Network modularization by using decomposition of Laplacian matrix

\[ \min_s s^t L s \]

- Use eigen decomposition principals:

\[ L \rightarrow (v_i, \lambda_i) \quad L = \sum_i \lambda_i v_i^t v_i \]

- Project \( s \) over eigenvectors of \( L \):

\[ s = \sum a_i v_i \]

\[ s^t L s = \sum a_i^2 \lambda_i \]

- Challenges in finding optimal \( a_i \)'s:
  - Without other conditions, a trivial solution exists
  - Second eigenvector characterizes partitioning
  - Vector \( s \) should be integer-valued => projection
Goals for today: Network analysis

1. Introduction to networks
   - Network types: regulatory, metab., signal., interact., func.
   - Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   - Local centrality metrics (degree, betweenness, closeness, etc)
   - Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   - Eigenvector and singular vector decomposition
   - Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   - Lasso and Elastic lasso
   - PCA and Sparse PCA

5. Network Communities and Modules
   - Guilt by association
   - Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   - Network diffusion kernels
   - Network deconvolution
Network Diffusion Kernels

• Define closeness of two nodes in the network

• One way: use weighted shortest path

• **Invariant** to the position of edges over a path
Definition of Diffusion Kernel

- A: Adjacency matrix,
- D: Diagonal matrix of Degrees
- L = D - A: Graph Laplacian Matrix
- **Diffusion kernel matrix** \( K = \exp(-\beta L) \)
  - \( \beta \): Diffusion parameter
- Matrix exponential, not element-wise exponential

**Intuition:** write Taylor series for matrices

\[
\exp(-\beta L) = \lim_{n \to \infty} \left(1 - \frac{\beta L}{n}\right)^n = \sum_{k=0}^{\infty} \frac{1}{k!}(-\beta L)^k
\]

(Kondor and Lafferty, 2002)
Adjacency Matrix and Degree Matrix

\[ A = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}, \quad D = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{pmatrix} \]
Graph Laplacian Matrix $L$

$L = D - A =$

$$
\begin{pmatrix}
1 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
-1 & -1 & 3 & -1 & 0 \\
0 & 0 & -1 & 2 & -1 \\
0 & 0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
$$
Interpretation via random walking

- Random walking according to transition probability
- Transition probability is constant $\beta$
- Remaining probability = Self loop

$$K_{ij}$$ is equal to the probability of the walk that started at $i$ being at $j$ after infinite time steps.
Interpretation via random walking

- Lazy random walk (Markov model):
- Transition probabilities:

\[ p(z_{l+1} = j | z_l = i) = \beta \quad \text{if} \quad i \sim j \]
\[ p(z_{l+1} = i | z_l = i) = 1 - d_i \beta \]

Transition matrix: \( T = I - \beta L \)

- After one step: \( p_1 = T p_0 \)
- After N steps: \( p_N = (I - \beta L)^N p_0 \)
- Reduce the time step 1 to \( \Delta t \)
- Diffusion parameter \( \beta \Delta t \)
- Taking the limit \( \Delta t \to 0 \)

\[ p_n = \lim_{n \to \infty} (I - \frac{\beta}{n} L)^n p_0 = e^{-\beta L} p_0 \]
Interpretation: Stochastic Process

• For each node \( i \), consider random variable \( Z_i(t) \)

• Initial condition
  – Zero mean, Variance
  – Independent to each other (covariance zero).

• Each variable sends a fraction to the neighbors

\[
Z_i(t + 1) = Z_i(t) + \beta \sum_{j \sim i} (Z_j(t) - Z_i(t))
\]

\[
Z(t + 1) = (I - \beta L)Z(t)
\]

\[
Z(t) = (I - \beta L)^t Z(0)
\]
Interpretation: Stochastic Process

• Time Evolution Operator \( T(t) = (I - \beta L)^t \)

• Covariance \( \text{Cov}_{ij}(t) = \sigma^2 T_{ij}(2t) \)

• Reduce the time step 1 to \( \Delta t \)
• Diffusion parameter \( \beta \Delta t \)
• Taking the limit \( \Delta t \to 0 \)

\[
\text{Cov}(t) = \sigma^2 \exp(-2\beta t L)
\]
Indirect information $\Rightarrow$ transitive edges

- **Correlation** and **mutual information** metrics lead to indirect edges in inferred networks due to transitivity $A \rightarrow B \rightarrow C$ leads to $A \rightarrow C$

- Indirect edges may be due to 2\textsuperscript{nd}, 3\textsuperscript{rd}, or **higher-order effects** (e.g. $A\rightarrow B\rightarrow C\rightarrow \ldots \rightarrow E$, etc)

- Each edge contains both **direct** and **indirect** components

**True network** ($G_{\text{dir}}$)

**Observed network** ($G_{\text{obs}}$)

**Transitive Effects**

Direct effects

Indirect effects
Model indirect effects as infinite sum of direct effects

\[ G_{\text{obs}} = G_{\text{dir}} + G_{\text{indir}} = G_{\text{dir}} + G_{\text{dir}}^2 + G_{\text{dir}}^3 + \ldots \]

Inverse operation: Network deconvolution

Given only \( G_{\text{obs}} \) can we find \( G_{\text{dir}} \)

Convergence condition: max eigenvalue \( |\lambda_{\text{max}}(G_{\text{dir}})| < 1 \)

- Infinite series converges when max absolute eigenvalue of direct network \( G_{\text{dir}} < 1 \). But \( G_{\text{dir}} \) is unknown!

- Key Ideas:
  1. scale observed network so that max absolute direct eigenvalue is <1
  2. express \( \lambda_{\text{dir}} \) as a (nonlinear) function of \( \lambda_{\text{obs}} \)
Scaling of $\lambda_{\text{dir}}$ to be $<\beta$ by scaling $\lambda_{\text{obs}}$

$$\lambda_{\text{dir}} = \left| \frac{\lambda_{\text{obs}}^{(us)}}{1/\alpha + \lambda_{\text{obs}}^{(us)}} \right|$$

- Let $\lambda^+_{\text{obs}}$ and $\lambda^-_{\text{obs}}$ be largest +/- eigenvalues of $G_{\text{obs}}$
- If we scale observed eigenvalues by:
  $$\alpha \leq \min \left( \frac{\beta}{(1 - \beta)\lambda^+_{\text{obs}}(us)}, \frac{-\beta}{(1 + \beta)\lambda^-_{\text{obs}}(us)} \right)$$
- Then max eigenvalue of direct network will be: $\beta$
- Intuitively: $\beta$ is the decay rate of indirect effects
  - $\beta$ small: rapid decay of indirect effects
  - $\beta$ large: indirect effects persist longer
- A priori selection
- Robustness analysis
- Cross validation
Network deconvolution framework

True network ($G_{\text{dir}}$)

Observed network ($G_{\text{obs}}$)

Transitive closure

Network deconvolution

Transitive closure:

Network deconvolution:

True network

$$G_{\text{dir}} = U \Sigma_{\text{dir}} U^{-1}$$

$$G_{\text{dir}} = G_{\text{obs}} (I + G_{\text{obs}})^{-1}$$

Observed network

$$G_{\text{obs}} = U \Sigma_{\text{obs}} U^{-1}$$

$$\lambda_i^{\text{dir}} = \frac{\lambda_i^{\text{obs}}}{1 + \lambda_i^{\text{obs}}}$$

Indirect effects

Series closed form

$$G_{\text{obs}} = G_{\text{dir}} + G_{\text{dir}}^2 + G_{\text{dir}}^3 + \ldots = G_{\text{dir}} (I - G_{\text{dir}})^{-1}$$

Feizi et al, Nature Biotech 2013
ND on regulatory networks (DREAM 5)

- **DREAM5**: Overa score improvement
- **Consistent** improvement as post-processing method.
- **New overall best performer** method

- Local network structure improvement
- Improve recovery of cascade motifs (specificity: no false feed-forward edges)
- High recovery feed-forward loops (sensitivity: recover true feed-forward edge)
Conclusion: Network analysis

1. Introduction to networks
   – Network types: regulatory, metab., signal., interact., func.
   – Bayesian (probabilistic) and Algebraic views

2. Network Centrality Measures
   – Local centrality metrics (degree, betweenness, closeness, etc)
   – Global centrality metrics (eigenvector centrality, page-rank)

3. Linear Algebra Review
   – Eigenvector and singular vector decomposition
   – Low rank approximations, Wigner semicircle law

4. Sparse Principal Component Analysis
   – Lasso and Elastic lasso
   – PCA and Sparse PCA

5. Network Communities and Modules
   – Guilt by association
   – Maximum cliques, density-based modules and spectral clustering

6. Network Diffusion Kernels and Deconvolution
   – Network diffusion kernels
   – Network deconvolution
Further reading

- Probabilistic graphical models
- Network Completion
- Non-negative matrix factorization
- Network Alignment
- Network Integration
Open issues

• Validation
  – How do we know the network structure is right?
• How do we know if the network function is right?
• Measuring and modeling protein expression
• Understanding the evolution of regulatory networks
Regulatory network: Input / output

**Trans** input: TF levels

**Cis** input: TF binding sites

Output: mRNA levels

- Gene expression prediction:

\[
G_i = f_i(G_1, G_2, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)
\]

Intractable to compute joint distribution

⇒ Focus on marginal distributions.
Very large number of regulators / targets

- Regulatory network limits the number of possible hypotheses
- Only directly related elements are connected
- Assume other pairs of nodes are conditionally independent
Predicting expression using regression trees

- Assumes variables are continuous. Arranges regulators in a tree
- Expression prediction follows a set of decision rules
  - Can model combinatorics
- Allows non-linear dependencies between regulators and target
- Targets can share regulatory programs

Expression of target modeled using Gaussians at each leaf node:

- $X_1 > e_1$
  - NO: Activating regulation
  - YES: Repressing regulation
- $X_2 > e_2$
  - NO: Activating regulation
  - YES: Repressing regulation

$N(\mu_{31}, \sigma_{31}) \quad N(\mu_{32}, \sigma_{32}) \quad N(\mu_{33}, \sigma_{33})$