But are the networks different?

Using Differential Network Analysis Software with Metabolite Data

COMETS Early Career Investigator Group Meeting: October 11, 2022

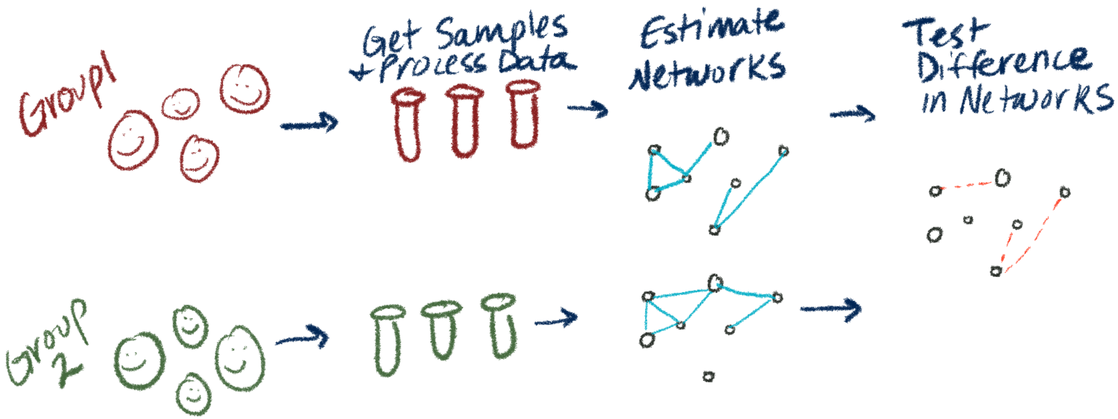
Margaret Janiczek

PhD Student, Biostatistics

University *of* Massachusetts Amherst

Motivation

- Identifying networks in biomedical data, and how they differ across populations, can help find drivers of disease and targets for treatment



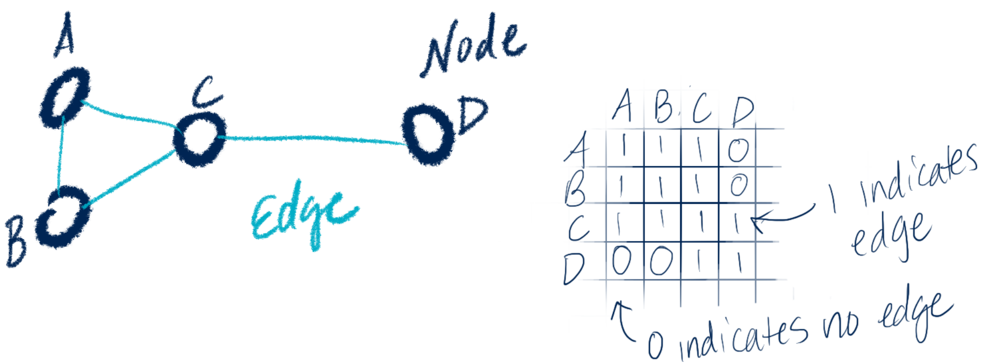
- Certain biomedical research questions lend themselves well to network/pathway analysis
    - Data from brain scans (Alzheimer's patient scans over time)
    - Gene expression (cancer vs normal tissue)
    - Microbiome (Crohn's disease vs Healthy Control)
    - Metabolomics - any applications from the group here?

Presentation Overview

- Background on graphical models and differential networks
- Overview of statistical landscape for differential network analysis
- Overview of available software
- Brief practical application using a few software options
- Discussion & feedback!

Background: Graphical Model

- Graphical models express connections between variables. When undirected, the connection doesn't imply any directionality.



- Connected edges can be seen in an **Adjacency Matrix**, where anything with a zero is considered "conditionally independent"
- In this example, A and B are **conditionally independent** of D

Gaussian Graphical Model

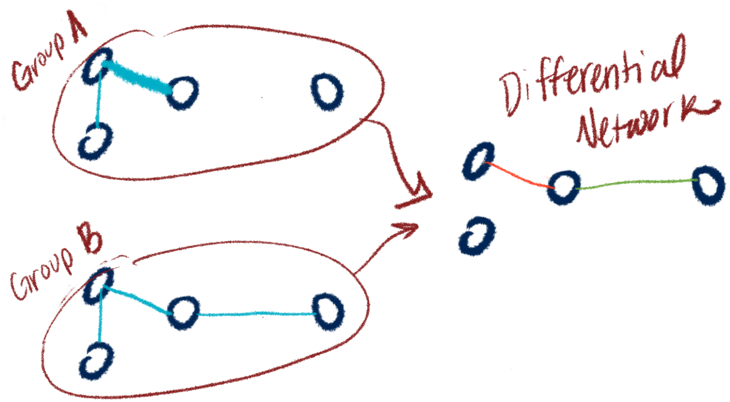
- Gaussian Graphical Models (GGMs) are the most widely used probabilistic graphical models (See Kate Shutta's recently published tutorial on GGMs for full details! [Shu+22])
- Assume data $\mathbf{x}$ is distributed as multivariate Gaussian $N(\mu, \Sigma)$ with mean vector μ and precision matrix $\Sigma^{-1} = \Theta$ whose entries correspond to partial correlation between variables
- So any two entries are conditionally independent if entry in Θ is zero.
- In low dimensional seeting, the Likelihood function:

$$l(\Theta; S) = ln|\Theta| - tr(S\Theta)$$

- Where $\Theta = \Sigma^{-1}$ is the "precision matrix" and S is sample covariance matrix.
- We want an estimate for this which we will call $P = S^{-1}$

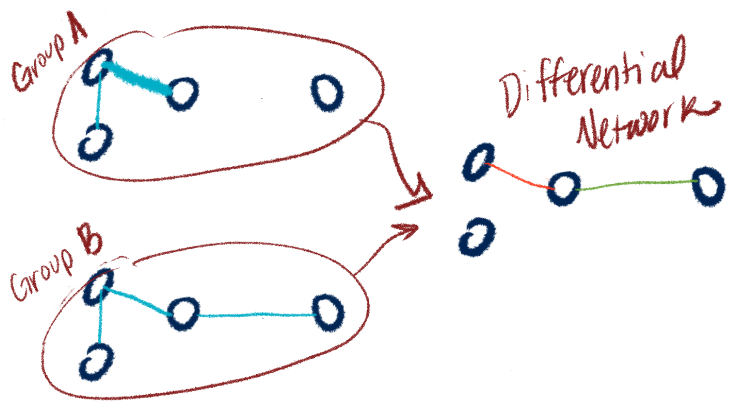
More than one graphical model

- Say you have data from two groups, like disease and healthy control.
- Say you estimate a graphical model for each group, then want to compare the resulting networks.



But are the networks different??

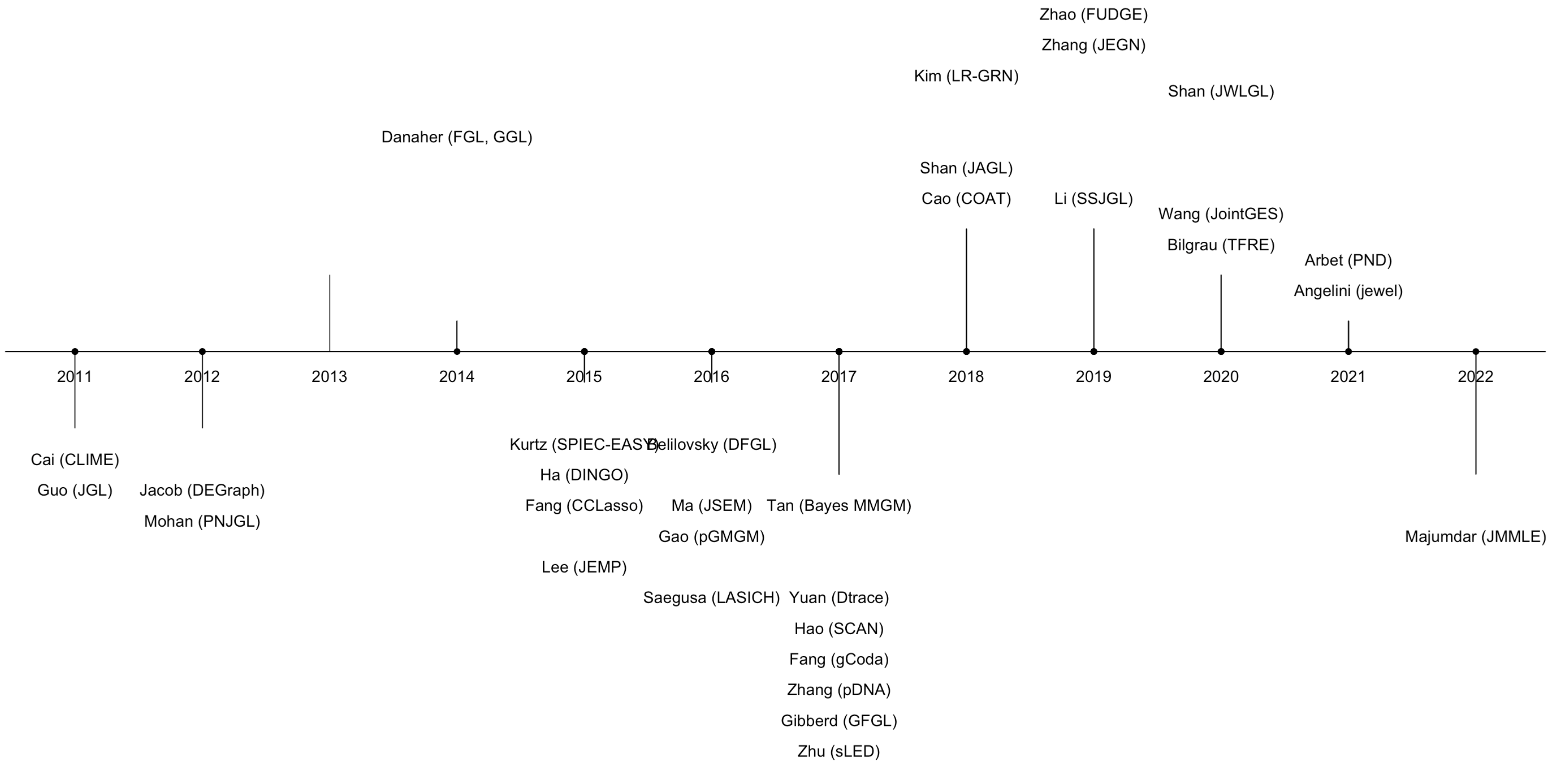
- How do you estimate them?
- How do you test the difference?
- How do you even *characterize* the difference? (edges? nodes? hubs? general structure?)
- This all falls under DIFFERENTIAL NETWORK ANALYSIS! (DiNA)



Statistical Landscape of DiNA methods

Timeline

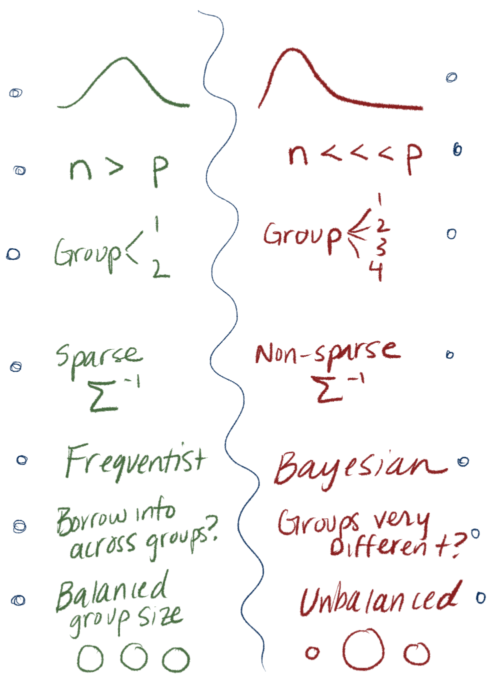
- I found 40+ methods papers on DiNA methods published in the last 10 years
- The wide variety is due to addressing many subtly different problems



Why so many methods?

To address various data and modeling situations!

What's your data like?



Other model considerations...

Hierarchical Structure

Cluster & GM

Latent Variables ???

Computation Time

Additional penalties ℓ_1 ℓ_2

Incorporate Known structure

DiNA Methods Summary

- Gaussian:
    - Graphical Lasso: JGL (Guo 2011)
        - Additional penalties for encouraging similar sparsity across groups: FGL & GGL (Danaher 2014)
            - Incorporating structural information: JSEM (Ma 2016)
                - Extension which doesn't require post-processing: jewel (Angelini 2021)
            - Doesn't require sparse inputs: DTrace (Yuan 2017)
                - Extension for multi-modal data: pDNA (Zhang 2017)
        - Node-based learning framework: PNJGL (Mohan 2014)
        - Unbalanced groups: JAGL (Shan 2018)
        - Hierarchical structure: JWLGL (Shan 2020)
    - Graphical Ridge: TFRE (Bilgrau 2020)
    - Adjust for global conditional dependencies to identify "driver" group-specific components: Dingo (Ha 2015)
    - Group-wise heterogeneous structure: LASICH (Saegusa 2016)
    - Direct estimation of difference: Zhao 2014
    - Uses latent nodes: Na 2019
    - Simulataneous clustering & GM estimation: SCAN (Hao 2017), Price 2021

DiNA Methods Summay (cont'd)

- Non-Gaussian: SPIEC-EASY (Kurtz 2015), pDNA (Zhang 2017)
- Semi-parametric: Xu 2016
- Comparison across 3+ groups: BioNetStat (Jardim 2019)
- Group-wise structure: JMMLE (Majumdar 2022)
- High dimensional: JointGES (Wang 2020), FUDGE (Zhao 2022)
- Bayesian: Peterson 2015, Mitra 2016, Tan 2017, Li 2019, Sekula 2022

Graphical Lasso (gLasso)

- Because majority of available methods are some variation on gLasso, I'm going to go into the details of the optimization problem and penalty terms here.
- Convex optimization problem for graphical lasso, where λ is a tuning parameter and $||\Theta||_1$ is the sum of absolute values of the elements of Θ. The solution gives an estimate for Σ^{-1}, the precision matrix:

$$maximize_{\Theta}\{logdet\Theta - tr(S\Theta) - \lambda||\Theta||_1\}$$

- *Graphical lasso* can be used even when $p >> n$, and when λ is large then it forces the estimated precision matrix to be sparse (so few edges!).
- Joint graphical lasso builds upon this by estimating *multiple, related GGMs* from data with observations belonging to distinct classes (for example, cancer vs normal tissue).
- The idea is to leverage information across the classes while still letting there be class-specific edges. Sparsity and similarity between graphs modified by penalty functions.

Notation

- K number of classes 2+. Index classes using k = 1, ... K.
- Σ_k^{-1}: True precision matrix for the kth class
- $Y^{(k)}$: n_k x p matrix consisting of n_k observations from the kth class on a set of p features which are common to all K datasets
- $S^{(k)}$: Empirical covariance matrix for $Y^{(k)}$
- $\Theta^{(k)}$: argument to convex optimization problem used for estimating Σ_k^{-1}
- Index matrix arguments by using i = 1, ..., p and j = 1, ..., p
- λ_1 and λ_2: non-negative tuning parameters used in penalty function

Major assumptions

- We assume the observations **within** each class are iid.
- Also assume μ_k, the mean for each class, is 0. i.e:

$$Y_1^{(k)}, \ldots, Y_{nk}^{(k)} \sim N(0, \Sigma_k)$$

Optimization problem for Joint Graphical Lasso

- Our goal is to estimate Σ_1^{-1}, ..., Σ_K^{-1} by using penalized log-likelihood approach.
- Again, we want each class to have it's own precision matrix, but to be able to use information across the classes to make them.
- Seek $\hat{\Theta}$ by solving:

$$maximize_{\{\Theta\}} \left(\sum_{k=1}^{K} n_k [log\{det(\Theta^{(k)})\} - tr(S^{(k)}\Theta^{(k)}) - P(\{\Theta\}) \right)$$

- A **major innovation of the Danaher 2014 paper**, is the generalization of the optimization problem to multiple classes, in addition to using the penalty function $P(\{\Theta\})$, for which the authors provide two different versions.

Penalty functions

- The general form for the penalty function is:

$$P(\{\Theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{ij}^{(k)}| + \widetilde{P}\{\Theta\}$$

- Notice that the $P(\{\Theta\})$ is **not class specific**. It takes information from all the classes!
- The form of this penalty function will encourage the solutions to share certain characteristics such as locations of sparsity or value.
- Depending on the form we choose and the value of the tuning parameters, we could essentially force joint graphical lasso to just perform unrelated graphical lasso on each of the K classes (i.e. if $\widetilde{P}\{\Theta\}$ is zero.)
- Let's look at the possible forms for $\widetilde{P}\{\Theta\}$!

Fused Graphical Lasso

- Fused Graphical Lasso (FGL) uses the following penalty function:

$$P(\{\Theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k')}|$$

- When λ_1 is **large**, FGL makes sparse estimates of $\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(K)}$
- When λ_2 is **large**, many elements of $\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(K)}$ will be the same across classes
- So, FGL "borrows information aggressively across classes, encouraging similar network structure and similar edge values"

Group Graphical Lasso

- Group Graphical Lasso (GGL) uses the following penalty function:

$$P(\{\Theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \left(\sum_{i,j} {\theta_{ij}^{(k)}}^2 \right)^{1/2}$$

- Lasso penalty applied to elements of the precision matrices
- Group lasso penalty is applied to the (i, j) element across all K precision matrices
- When λ_1 is **large**, GGL makes sparse estimates of $\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(K)}$
- So, GGL just encourages a shared pattern of *sparsity*, not shared *edge values* (unlike FGL which encourage sharing across both)

Other penalty functions

- Here is a selection of penalty functions for comparison. For full treatment see Tsai or Shojaie review papers.

The OG Joint Graphical Lasso

JGL $\lambda_1 \sum_{i \neq j} \theta_{i,j} + \lambda_2 \sum_{k=1}^{K} \sum_{i \neq j} |\gamma_{i,j}^{(k)}|$

Important ☆ Danaher 2014

FGL $\lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{i,j}^{k}| + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{i,j}^{(k)} - \theta_{i,j}^{(k')}|$

GGL $\lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{i,j}^{(k)}| + \lambda_2 \sum_{i \neq j} \left(\sum_{i,j} \theta_{i,j}^{(k)2} \right)^{1/2}$

LASICH $\lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{i,j}^{(k)}| + \lambda_1 \lambda_2 \sum_{i \neq j} \left\{ \sum_{k,k'}^{K} W_{k,k'} \left(\theta_{i,j}^{(k)} + \theta_{i,j}^{(k')} \right)^2 \right\}^{1/2}$

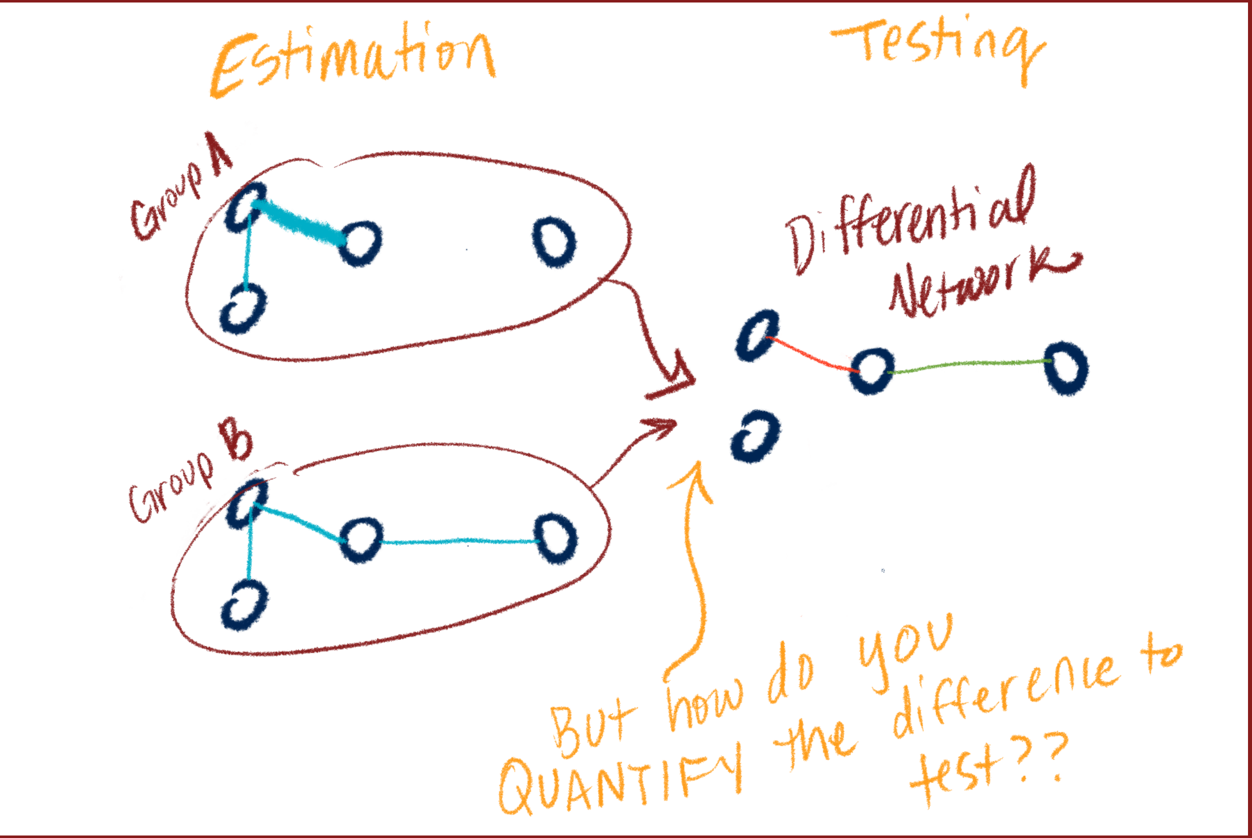
Group-wise heterogeneous structure

JSEM $\sum_{j \neq i} \sum_{g \in G_{i,j}} \lambda_{ij}^{(g)} \|\theta_{i,j}^{(g)}\|_2$

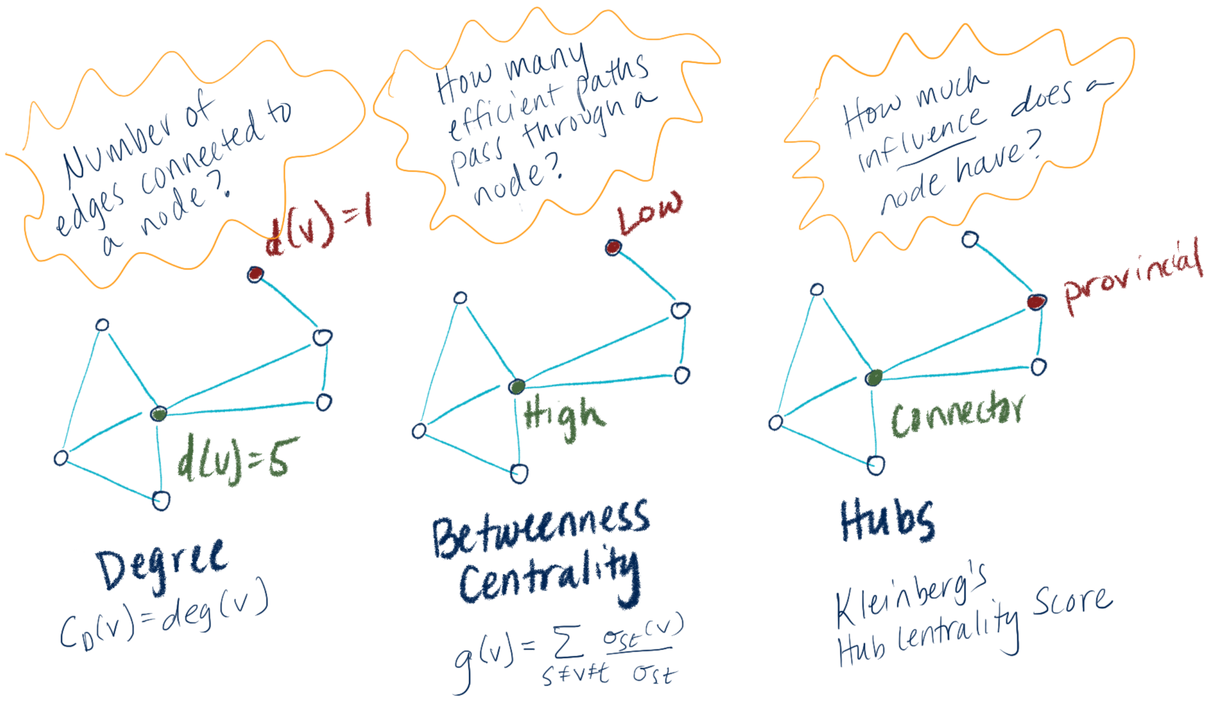
Neighborhood selection

JAGL $\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \neq j} \frac{1}{\underbrace{|(1-\pi)\hat{t}_{i,j} + \pi \hat{s}_{i,j}|^{r}}_{\text{weight avg of pooled and individual precision matrices}}} |\theta_{i,j}^{(k)}|$

For K unbalanced groups


Estimation
Testing
Group A
Group B
Differential Network
But how do you QUANTIFY the difference to test??

Some Node Importance Measures



And finally… test the difference

Methods & Software for testing

- Lichtblau 2017 Compares 10 methods for quantiyfing node-specific differences between groups
- Identify pairs of nodes with difference (Ha DINGO 2015, McKenzie DGCA 2016)
- Identify subsets (3+) nodes that whose connections are different between groups (Jardim BioNetStat 2019, Arbet PND 2021)

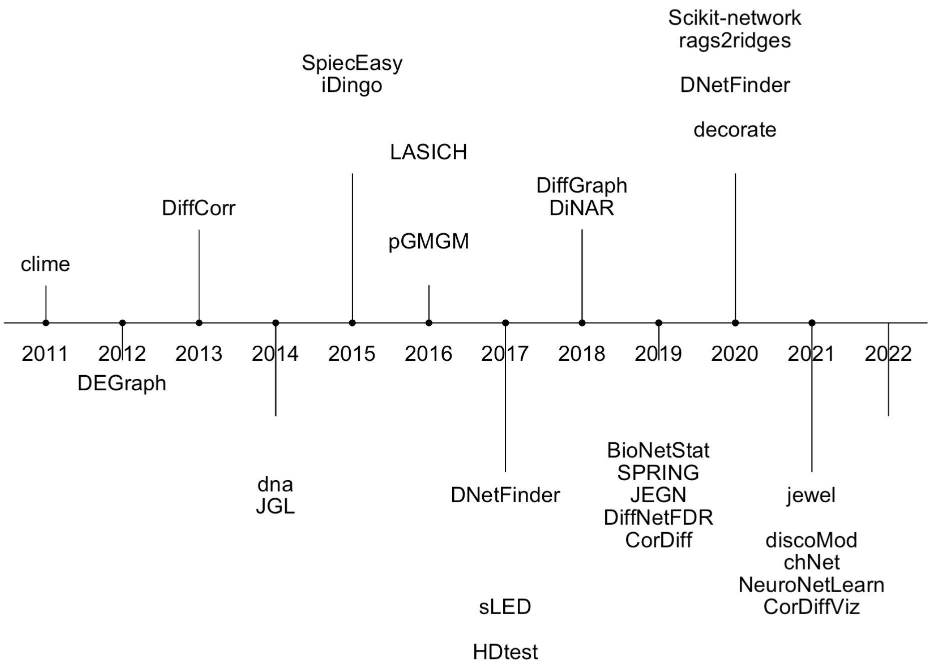
- Various p-value options, e.g. permutation

- Adjust for multiple testing!! Bonferoni for conservative estimate, FDR for less stringent.

Software Landscape of DiNA methods

Overview DiNA software landscape

- I found 26 different R packages and 2 Python packages that implement a variety of subtly different DiNA algorithms/pipelines



Notes on software

- `JGL`, `iDingo`, `rags2ridges`, and `SpiecEasy` seem to be most popular and cited.
- I have a full tutorial for `JGL` posted on my GitHub, and Kate has one available for `iDingo`.
- Currently working making tutorials for for `rags2ridges`, `Spiec-Easy` and will work through the other available methods

JGL

- JGL package runs Fused Graphical Lasso (FGL) and Group Graphical Lasso from Danaher et al 2014
- Estimates sparse covariance matrices that are *similar* across classes
- Has a lot of useful functions to analyze the networks after estimating them, for example extracting hubs, edges, degree etc.
- Graphical lasso uses L1 penalty, which encourages sparsity and as a result selects edges in the graph in the process of estimating precision matrix
- (If time allows we can open up my JGL tutorial for a practical metabolomics example)

Takeaways

- DiNA has potential to be a useful tool in biomedical research
- There are many ways to customize the estimation and testing process to fit research question and data types
- However the broad landscape of methods and software and the current lack of practical applied tutorials comparing software methods seems like a barrier to widespread use
- I'm working on trying to bridge the gap between statistical methodology and applied researchers! Full tutorial forthcoming!

Questions & Comments?

Thank you!

- Dr. Raji Balasubramanian & Balasubramanian Lab
- Dr. Kate Hoff Shutta

Github: @mljaniczek
Website: mljaniczek.github.io/
Slides created via the R package **xaringan**.

References

Cai, T., W. Liu, and X. Luo (2011). "A Constrained ℓ1 Minimization Approach to Sparse Precision Matrix Estimation". In: *Journal of the American Statistical Association* 106.494. Publisher: Taylor & Francis _ eprint: https://doi.org/10.1198/jasa.2011.tm10155, pp. 594-607. ISSN: 0162-1459. DOI: 10.1198/jasa.2011.tm10155. URL: https://doi.org/10.1198/jasa.2011.tm10155 (visited on Aug. 18, 2022).

Danaher, P., P. Wang, and D. M. Witten (2014). "The joint graphical lasso for inverse covariance estimation across multiple classes". In: *J R Stat Soc Series B Stat Methodol* 76.2, pp. 373-397. ISSN: 1369-7412. DOI: 10.1111/rssb.12033.

Guo, J., E. Levina, G. Michailidis, et al. (2011). "Joint estimation of multiple graphical models". In: *Biometrika* 98.1, pp. 1-15. ISSN: 0006-3444. DOI: 10.1093/biomet/asq060.

Jacob, L., P. Neuvial, and S. Dudoit (2012). "More power via graph-structured tests for differential expression of gene networks". In: *The Annals of Applied Statistics* 6.2. Publisher: Institute of Mathematical Statistics, pp. 561-600. ISSN: 1932-6157, 1941-7330. DOI: 10.1214/11-AOAS528. URL: https://projecteuclid.org/journals/annals-of-