

# Landscape of Differential Network Analysis Methods & Software

**Margaret Janiczek**

PhD Student, Biostatistics

University of  
Massachusetts  
Amherst

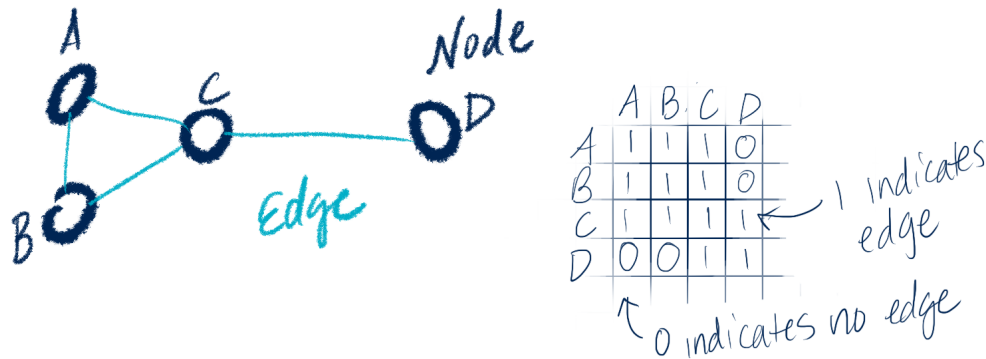


# Presentation Overview

- Background on graphical models and differential networks
- Overview of statistical landscape for differential network analysis
- Overview of available software
- Links to 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", and anything with a 1 is considered "conditionally dependent"
- In this example, A and B are **conditionally independent** of D

# Gaussian Graphical Model

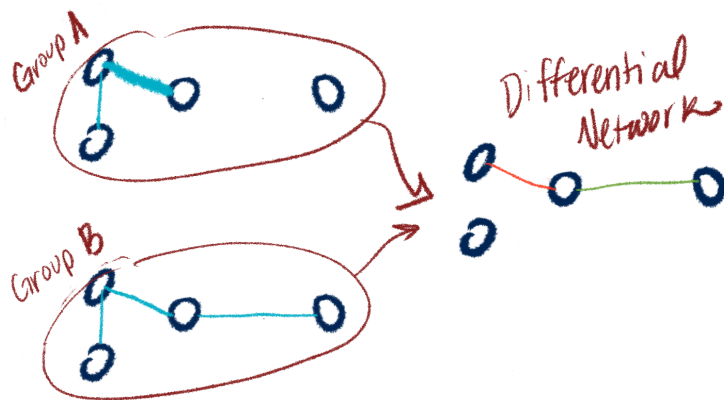
- Gaussian Graphical Models (GGMs) are the most widely used probabilistic graphical models (See Kate Shutta's recently published tutorial on GGMs for full details! [Shu+22])
- Assume data  $\mathbf{y}$  is distributed as multivariate Gaussian  $N(\mu, \Sigma)$  with mean vector  $\mu$  and precision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

$$l(\Omega; S) = \ln|\Omega| - \text{tr}(S\Omega)$$

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

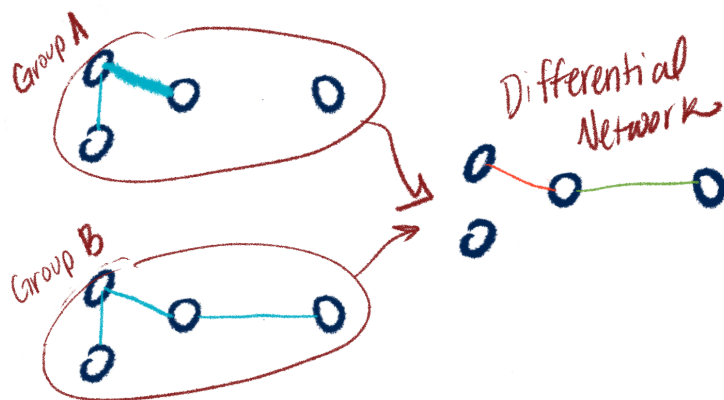
# More than one graphical model

- 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)

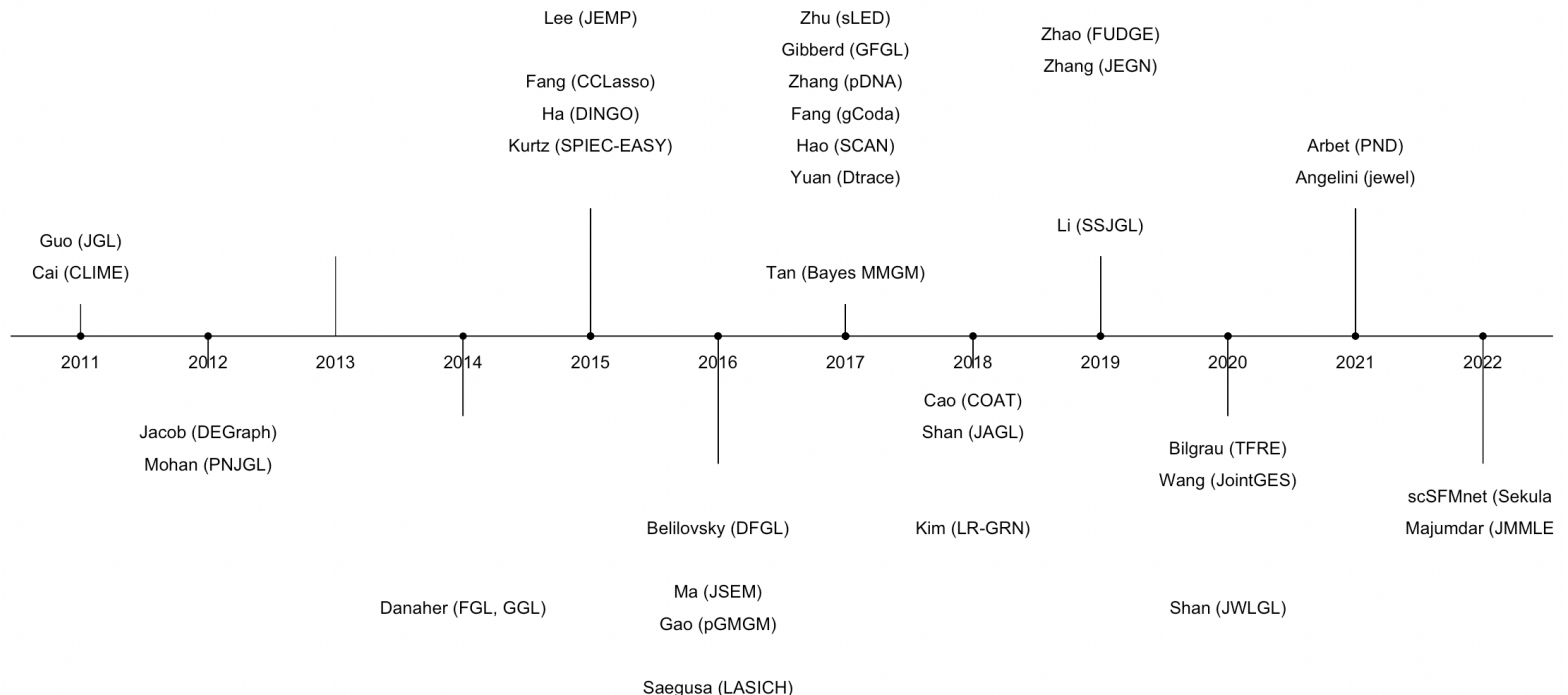


# Statistical Landscape of DiNA methods



# Timeline


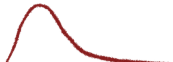
- I found 40+ methods papers on DiNA methods published in the last 10 years
- The wide variety is due to addressing many subtly different problems



# Why so many methods?

To address various data and modeling situations!

What's your data like?

- |  |   |
|--|---|
|   |    |
| <ul style="list-style-type: none"> <li>◦ <math>n &gt; p</math></li> <li>◦ <math>\text{Group} &lt; \frac{1}{2}</math></li> <li>◦ Sparse <math>\Sigma^{-1}</math></li> <li>◦ Frequentist</li> <li>◦ Borrow info across groups?</li> <li>◦ Balanced group size<br/>○ ○ ○</li> </ul> | <ul style="list-style-type: none"> <li>◦ <math>n \ll p</math></li> <li>◦ <math>\text{Group} &lt; \frac{1}{2}</math><br/>1<br/>2<br/>3<br/>4</li> <li>◦ Non-sparse <math>\Sigma^{-1}</math></li> <li>◦ Bayesian</li> <li>◦ Groups very different?</li> <li>◦ Unbalanced<br/>○ ○ ○</li> </ul> |

Other model considerations...

Hierarchical Structure  $\begin{matrix} \text{---} \\ \text{---} \end{matrix}$

Cluster & GM  $\begin{matrix} \circ & \circ & \circ \\ \circ & \circ & \circ \end{matrix}$

Latent Variables?  $\begin{matrix} \text{---} \\ \text{---} \end{matrix}$

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
  - Simultaneous 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  $\lambda$  is a tuning parameter and  $\|\Theta\|_1$  is the sum of absolute values of the elements of  $\Theta$ . The solution gives an estimate for  $\Sigma^{-1}$ , the precision matrix:

$$\underset{\Theta}{\text{maximize}} \{ \log \det \Theta - \text{tr}(S\Theta) - \lambda \|\Theta\|_1 \}$$

- *Graphical lasso* can be used even when  $p \gg n$ , and when  $\lambda$  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  $k = 1, \dots, K$ .
- $\Sigma_k^{-1}$ : True precision matrix for the  $k$ th class
- $Y^{(k)}$ :  $n_k \times p$  matrix consisting of  $n_k$  observations from the  $k$ th class on a set of  $p$  features which are common to all  $K$  data sets
- $S^{(k)}$ : Empirical covariance matrix for  $Y^{(k)}$
- $\Theta^{(k)}$ : argument to convex optimization problem used for estimating  $\Sigma_k^{-1}$
- Index matrix arguments by using  $i = 1, \dots, p$  and  $j = 1, \dots, p$
- $\lambda_1$  and  $\lambda_2$ : non-negative tuning parameters used in penalty function

# Major assumptions

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

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

# Optimization problem for Joint Graphical Lasso

- Our goal is to estimate  $\Sigma_1^{-1}, \dots, \Sigma_K^{-1}$  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:

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

- 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:

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

- 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  $\tilde{P}\{\Theta\}$  is zero.)
- Let's look at the possible forms for  $\tilde{P}\{\Theta\}$ !

# 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  $\lambda_1$  is **large**, FGL makes sparse estimates of  $\hat{\Theta}^{(1)}, \dots, \hat{\Theta}^{(K)}$
- When  $\lambda_2$  is **large**, many elements of  $\hat{\Theta}^{(1)}, \dots, \hat{\Theta}^{(K)}$  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  $\lambda_1$  is **large**, GGL makes sparse estimates of  $\hat{\Theta}^{(1)}, \dots, \hat{\Theta}^{(K)}$
- 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  
 Important ★  
 Danaher 2014

JGL  $\lambda_1 \sum_{i \neq j} \theta_{ij} + \lambda_2 \sum_{k=1}^K \sum_{i \neq j} |\gamma_{ij}^{(k)}|$   
 FGL  $\lambda_1 \sum_{i \neq j} \sum_{k=1}^K |\theta_{ij}^{(k)}| + \lambda_2 \sum_{k \neq k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k')}|$   
 GGL  $\lambda_1 \sum_{k=1}^K \sum_{i \neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} (\sum_{j} \theta_{ij}^{(k)})^2$

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

JSEM  $\sum_{j \neq i} \sum_{g \in \mathcal{G}_{ij}} \lambda_{ij}^{(g)} \|\theta_{ij}^{(g)}\|_2$   
 ↳ Neighborhood selection

JAGL  $\sum_{k=1}^K \frac{1}{n_k} \sum_{i \neq j} \underbrace{\left( \frac{1}{(1-\pi)} \mathbb{E}_{i,j} + \pi \hat{\mathbb{S}}_{i,j} \right)^r}_{\text{weight avg of pooled and individual precision matrices}} |\theta_{ij}^{(k)}|$   
 ↳ For K unbalanced groups

# Bayesian Inference of Multiple GGMs

- Peterson, Stingo, and Vannucci (2015) offer an alternative Bayesian approach.
- They use Markov random field (MRF) prior which encourages common structures across groups, but does not assume that all subgroups are related
- Place a spike-and-slab prior on parameters that measure network relatedness in an effort to learn which groups have a shared graph structure
- Posterior probabilities of inclusion for those parameters summarize the network similarity.

# Intuition behind spike-and-slab prior

# Intuition behind spike-and-slab prior

# Notation

- $K$  number of classes 2+. Index classes using  $k = 1, \dots, K$ .
- $\Sigma_k^{-1}$ : True precision matrix for the  $k$ th class
- $Y^{(k)}$ :  $n_k \times p$  matrix consisting of  $n_k$  observations from the  $k$ th class on a set of  $p$  features which are common to all  $K$  data sets.  $n_k$  need not be identical size.
- $S^{(k)}$ : Empirical covariance matrix for  $Y^{(k)}$
- $\Omega^{(k)}$ : symmetric positive definite matrix constrained by a graph  $G_k$
- $\mathbf{g}_{ij}$ : Binary vector representing inclusion of edge  $(i, j)$  in graphs 1, ...,  $K$ .
- Index matrix arguments by using  $i = 1, \dots, p$  and  $j = 1, \dots, p$



# Markov Random Field Prior

Probability of the binary vector of edge inclusion indicators  $\mathbf{g}_{ij}$  given by:

$$p(\mathbf{g}_{ij}|v_{ij}, \Theta) = C(v_{ij}, \Theta)^{-1} \exp(v_{ij} \mathbf{1}^T \mathbf{g}_{ij} + \mathbf{g}_{ij}^T \Theta \mathbf{g}_{ij})$$

- $\mathbf{1}$ : unit vector of dimension K
- $v_{ij}$  is a parameter specific to each set of edges  $\mathbf{g}_{ij}$
- $\Theta$  is a KxK symmetric matrix representing pairwise relatedness of the graphs for each sample group.
  - Diagonals are zero, off-diagonals which are non-zero represent connections between related networks.
- $C(v_{ij}, \Theta)$ : Normalizing constant

# MRF continued

So, prior probability that edge (i, j) is absent from all K graphs is:

$$p(\mathbf{g}_{ij} = 0 | v_{ij}, \Theta) = 1/C(v_{ij}, \Theta)$$

- Joint prior on graphs is product of densities for each edge:

$$p(G_1, \dots, G_K | v, \Theta) = \prod_{i < j} p(\mathbf{g}_{ij} = 0 | v_{ij}, \Theta)$$

Conditional probability of the inclusion of edge in  $G_k$ , given inclusion of edge in remaining graphs, is:

$$p(g_{k,ij} | g_{m,ij} \text{ for } m \neq k, v_{ij}, \Theta) = \frac{\exp(g_{k,ij}(v_{ij} + 2 \sum_{m \neq k} \theta_{km} g_{m,ij}))}{1 + \exp(v_{ij} + 2 \sum_{m \neq k} \theta_{km} g_{m,ij})}$$

# Selection prior on network similarity

Impose priors on  $v$  and  $\Theta$  to reduce false selection of edges.

- Use spike-and-slab prior on the off-diagonal entries of  $\Theta$ :  $\theta_{km}$ 
  - "Slab" portion defined on positive domain
  - Appropriate choice is  $\text{Gamma}(x|\alpha, \beta)$  with  $\alpha > 1$
  - So mixture prior on  $\theta_{km}$  is  $p(\theta_{km}|\gamma_{km})$
  - Joint prior on off-diagonal entries is product of marginal densities:
  - $p(\Theta|\gamma) = \prod_{k < m} p(\theta_{km}|\gamma_{km})$
  - Place Bernoulli prior on latent indicators

# Edge-specific informative prior on $v$

- Given a known reference network  $G_0$ , define a prior that encourages higher selection probabilities for edges included in  $G_0$ .
- Impose a prior on probability of inclusion in edge in  $G_k$  which reflects belief that it is similar to the reference network. Use Beta(a,b) prior.
- In cases where no prior knowledge on graph structure, use a prior that favors lower values such as Beta(1, 4) to encourage overall sparsity.

# Joint Posterior

- Joint posterior given  $\Psi$  is set of all parameters and  $X$  is observed data for all groups:

# MCMC Sampling

- Since posterior distribution is analytically intractable, construct Markov chain Monte Carlo (MCMC) sampler to obtain a posterior sample of the parameters of interest.
  - At iteration  $t$ :
  - Update graph  $G_k^{(t)}$  and precision matrix  $\Omega_k^{(t)}$  for each group  $k = 1, \dots, K$
  - Update parameters for network relatedness  $\theta_{km}^t$  and  $\gamma_{km}^t$  for  $1 \leq k < m \leq K$
  - Update edge-specific parameters  $v_{ij}^{(t)}$  for  $1 \leq i < j \leq p$

# Other Bayesian methods for inferring multiple GGMs

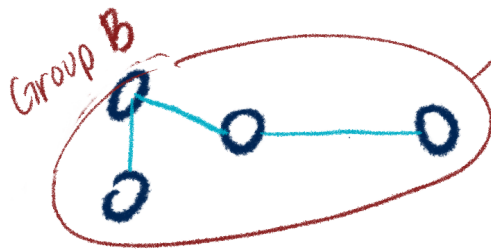
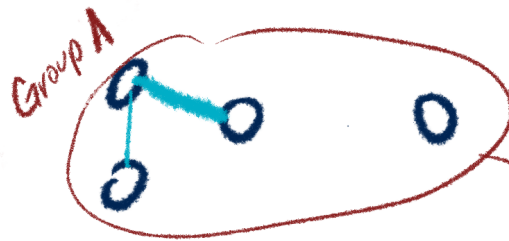
- Method for non-normal and mixed discrete continuous 'omic data (Bhadra 2018)
- Bayesian counterpart of JGL with simultaneous shrinkage and model selection (Li 2019)
- Multi-layered genomic networks - good for when you have multiple data types/hierarchical structure (Ha 2020)
- Hierarchical Bayesian factor model for count data (good for single-cell differential network analysis) (Sekula 2021)

# Takeaway from methods landscape

- There are many available methods to jointly estimate multiple graphical models
- Choice of method depends on your data type, goal of analysis, and ease of implementation (more on that in software section)



Estimation

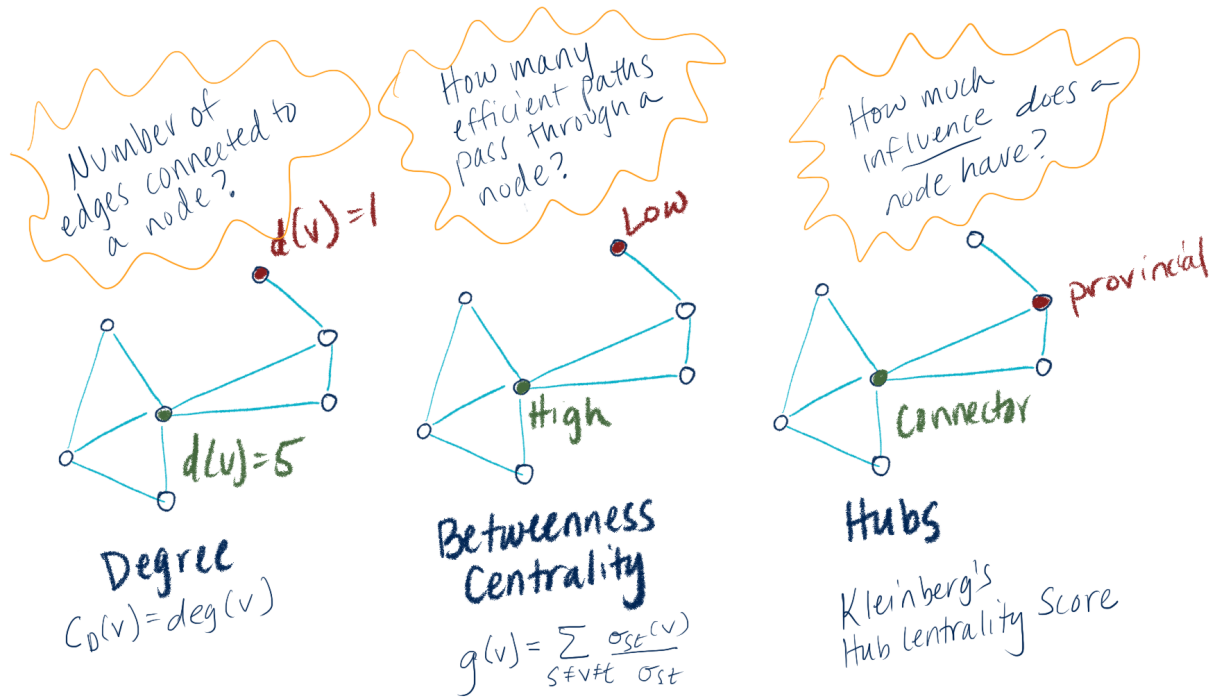


Testing



BUT how do you  
QUANTIFY the difference to  
test??

# Some Node Importance Measures



**And finally... test the difference**

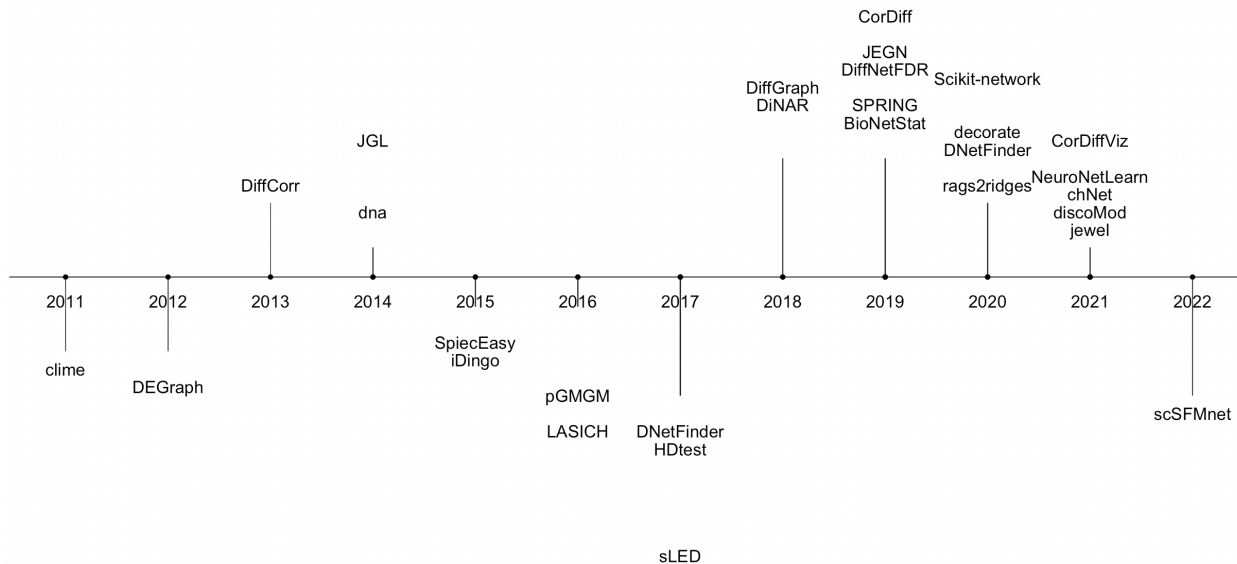
# Methods & Software for testing

- Lichtblau 2017 Compares 10 methods for quantifying 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.
- For Bayesian methods: posterior mean and  $100(1 - \alpha)\%$  Credible Interval for each gene-gene pair correlation difference are obtained from the posterior. (Sekula 2022)

# 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
- For implementation in Python, see my repository: [https://github.com/mljaniczek/diff\\_net\\_python](https://github.com/mljaniczek/diff_net_python) which contains scripts on running sparse inverse covariance estimation methods in Python (Graphical Lasso and Ledoit-Wolf shrinkage methods).
- Bayesian methods: I am working on the spikeyglass package which implements Li's 2019 method for Bayesian Joint Spike-and-Slab Graphical Lasso. The scSFMnet package is also available for Sekula 2022 method for hierarchical Bayesian factor model, which can be used on zero-inflated count data.

# 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
- Members of **Balasubramanian Lab**
- Dr. Kate Hoff Shutta
- Dr. Zehang Richard Li

Github: @mljaniczek

Website: [mljaniczek.github.io/](https://mljaniczek.github.io/)

Slides created via the R package **xaringan**.

# References

Cai, T., W. Liu, and X. Luo (2011). "A Constrained  $\ell_1$  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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1214/11-AOAS528). URL: <https://projecteuclid.org/journals/annals-of->