

Statistical and computational methods for bioinformatics and social network analysis

or how did I learn to stop worrying and love the bomb

George G Vega Yon

University of Southern California, Department of Preventive Medicine

October 4, 2019

Paper 1: Exponential Random Graph Models for Small Networks

Paper 2: On the prediction of gene functions using phylogenetic trees

Future directions

Things that are very interesting but I most probably won't have any time to discuss with the attendees

References

Paper 1: Exponential Random Graph Models for Small Networks

Paper 2: On the prediction of gene functions using phylogenetic trees

Future directions

Things that are very interesting but I most probably won't have any time to discuss with the attendees

References

What are Exponential Random Graph Models

Exponential Family Random Graph Models, aka ERGMs are:

What are Exponential Random Graph Models

Exponential Family Random Graph Models, aka ERGMs are:

- ▶ Statistical models of (social) networks

What are Exponential Random Graph Models

Exponential Family Random Graph Models, aka ERGMs are:

- ▶ Statistical models of (social) networks
- ▶ In simple terms: statistical inference on what network patterns/structures/motifs govern the data-generating process

What are Exponential Random Graph Models

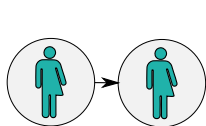
Exponential Family Random Graph Models, aka ERGMs are:

- ▶ Statistical models of (social) networks
- ▶ In simple terms: statistical inference on what network patterns/structures/motifs govern the data-generating process

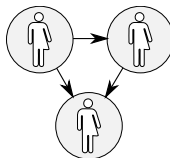
What are Exponential Random Graph Models

Exponential Family Random Graph Models, aka ERGMs are:

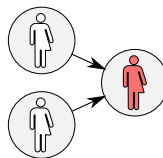
- ▶ Statistical models of (social) networks
- ▶ In simple terms: statistical inference on what network patterns/structures/motifs govern the data-generating process



Homophily



Transitive Triad



Popularity

A vector of
model parameters

A vector of
sufficient statistics




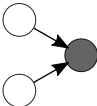

$$\Pr(\mathbf{Y} = \mathbf{y} \mid \theta, \mathbf{X}) = \frac{\exp\{\theta^t \mathbf{s}(\mathbf{y}, \mathbf{X})\}}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\{\theta^t \mathbf{s}(\mathbf{y}', \mathbf{X})\}}, \quad \forall \mathbf{y} \in \mathcal{Y}$$

Observed data

The normalizing
constant

All possible
networks

Sufficient statistics have various forms

Representation	Description
	Mutual Ties (Reciprocity) $\sum_{i \neq j} y_{ij} y_{ji}$
	Transitive Triad (Balance) $\sum_{i \neq j \neq k} y_{ij} y_{jk} y_{ik}$
	Homophily $\sum_{i \neq j} y_{ij} \mathbf{1}(x_i = x_j)$
	Covariate Effect for Incoming Ties $\sum_{i \neq j} y_{ij} x_j$
	Four Cycle $\sum_{i \neq j \neq k \neq l} y_{ij} y_{jk} y_{kl} y_{li}$

Small-large (dozens to a couple of thousand vertices) networks

- ▶ Markov Chain Monte Carlo (MCMC) based approaches like MC-MLE or Robbins-Monro Stochastic approximation. [▶ details](#)
- ▶ Maximum Pseudo Likelihood (MPLE)

Small-large (dozens to a couple of thousand vertices) networks

- ▶ Markov Chain Monte Carlo (MCMC) based approaches like MC-MLE or Robbins-Monro Stochastic approximation. [▶ details](#)
- ▶ Maximum Pseudo Likelihood (MPLE)

large-huge networks (up to the millions of vertices)

- ▶ Semi-parametric bootstrap
- ▶ Conditional joint estimation (like snowball sampling, a.k.a. divide and conquer)
- ▶ Equilibrium Expectation Algorithm (millions of vertices)

Small-large (dozens to a couple of thousand vertices) networks

- ▶ Markov Chain Monte Carlo (MCMC) based approaches like MC-MLE or Robbins-Monro Stochastic approximation. [▶ details](#)
- ▶ Maximum Pseudo Likelihood (MPLE)

large-huge networks (up to the millions of vertices)

- ▶ Semi-parametric bootstrap
- ▶ Conditional joint estimation (like snowball sampling, a.k.a. divide and conquer)
- ▶ Equilibrium Expectation Algorithm (millions of vertices)

What about tiny to small networks?

MC-MLE works great (we have some simulations showing this), but it has some problems:

MC-MLE works great (we have some simulations showing this), but it has some problems:

- Possible accuracy issues (error rates)

What shall we do then?

MC-MLE works great (we have some simulations showing this), but it has some problems:

- ▶ Possible accuracy issues (error rates)
- ▶ Prone to degeneracy problems (sampling and existence of MLE)

What shall we do then?

MC-MLE works great (we have some simulations showing this), but it has some problems:

- ▶ Possible accuracy issues (error rates)
- ▶ Prone to degeneracy problems (sampling and existence of MLE)
- ▶ It is not MLE,

What shall we do then?

A frame with examples of small networks...



ERGMs for Small Networks

- In the case of small-enough networks, computation of the likelihood becomes computationally feasible.

- ▶ In the case of small-enough networks, computation of the likelihood becomes computationally feasible.
- ▶ For example, a network with 5 nodes has 1,048,576 unique configurations.

- ▶ In the case of small-enough networks, computation of the likelihood becomes computationally feasible.
- ▶ For example, a network with 5 nodes has 1,048,576 unique configurations.
- ▶ This allow us to directly compute **the normalizing constant**.

- ▶ In the case of small-enough networks, computation of the likelihood becomes computationally feasible.
- ▶ For example, a network with 5 nodes has 1,048,576 unique configurations.
- ▶ This allow us to directly compute **the normalizing constant**.
- ▶ Using the exact likelihood opens a huge window of methodological-possibilities.

- ▶ In the case of small-enough networks, computation of the likelihood becomes computationally feasible.
- ▶ For example, a network with 5 nodes has 1,048,576 unique configurations.
- ▶ This allow us to directly compute **the normalizing constant**.
- ▶ Using the exact likelihood opens a huge window of methodological-possibilities.

- ▶ In the case of small-enough networks, computation of the likelihood becomes computationally feasible.
- ▶ For example, a network with 5 nodes has 1,048,576 unique configurations.
- ▶ This allow us to directly compute **the normalizing constant**.
- ▶ Using the exact likelihood opens a huge window of methodological-possibilities.

We implemented this and more in the `ergmito` R package [▶ more](#)

In order to compare the MLE with the MC-MLE estimation method, we performed a simulation study with the following features:

In order to compare the MLE with the MC-MLE estimation method, we performed a simulation study with the following features:

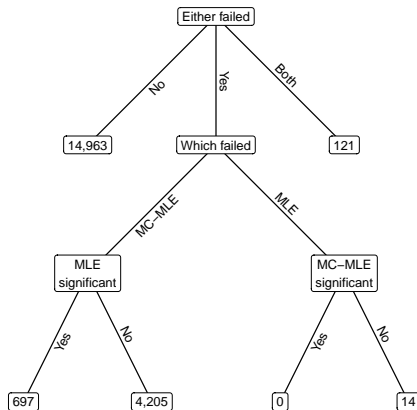
- ▶ Draw 20,000 samples of groups of small networks

In order to compare the MLE with the MC-MLE estimation method, we performed a simulation study with the following features:

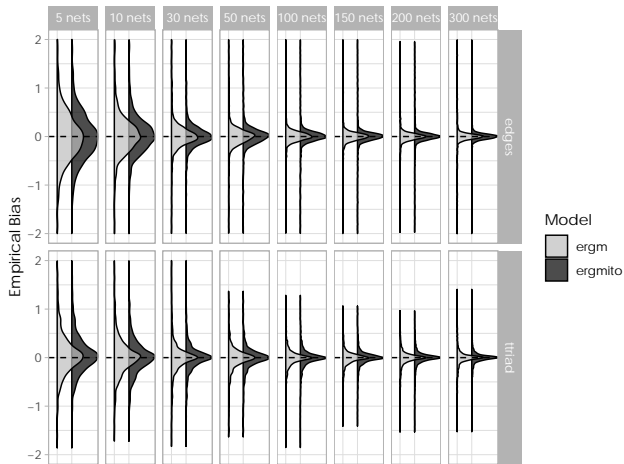
- ▶ Draw 20,000 samples of groups of small networks
- ▶ Each group had prescribed: (model parameters, number of networks, sizes of the networks)

In order to compare the MLE with the MC-MLE estimation method, we performed a simulation study with the following features:

- ▶ Draw 20,000 samples of groups of small networks
- ▶ Each group had prescribed: (model parameters, number of networks, sizes of the networks)
- ▶ We estimated the models using the `ergm` and `ergmito` R packages



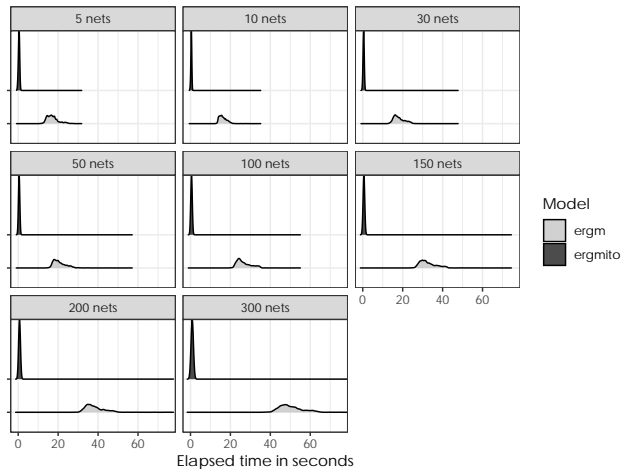
Paper 1 Simulation Studies: Empirical Bias



Sample size	N. Simulations	P(Type I error)		
		MC-MLE	MLE	chi2
5	2,189	0.084	0.057	11.71 ***
10	2,330	0.070	0.045	12.46 ***
15	2,395	0.084	0.066	5.55 *
20	2,430	0.074	0.060	3.58
30	2,460	0.057	0.052	0.67
50	2,495	0.046	0.044	0.17
100	2,499	0.048	0.048	0.00

Table 1 Empirical Type I error rates. The χ^2 statistic is from a 2-sample test for equality of proportions, and the significance levels are given by *** $p < 0.001$, ** $p < 0.01$, and * $p < 0.05$. The lack of fitted samples in some levels is due to failure of the estimation method.

Paper 1 Simulation Studies: Elapsed time



Paper 1: Key takeaway

- Developed a new extension of ERGMs using exact statistics for small networks (families, teams, ego-centered, etc.)

- ▶ Developed a new extension of ERGMs using exact statistics for small networks (families, teams, ego-centered, etc.)
- ▶ Performance: Same (un)bias, Lower Type I error rates, (way) faster.

- ▶ Developed a new extension of ERGMs using exact statistics for small networks (families, teams, ego-centered, etc.)
- ▶ Performance: Same (un)bias, Lower Type I error rates, (way) faster.
- ▶ Opens the door the new methods.

Paper 1: Exponential Random Graph Models for Small Networks

Paper 2: On the prediction of gene functions using phylogenetic trees

Future directions

Things that are very interesting but I most probably won't have any time to discuss with the attendees

References

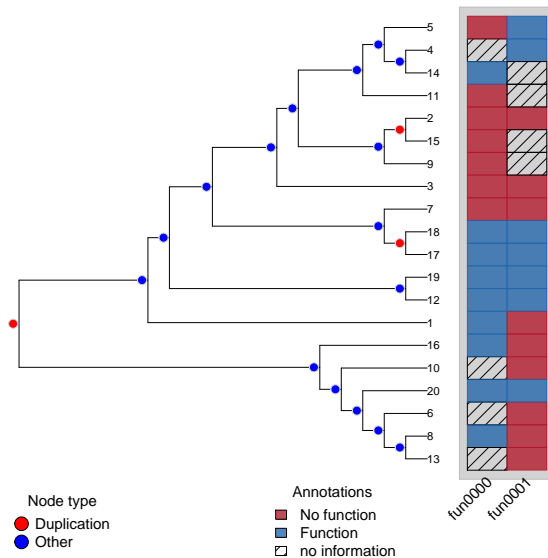
Phylogenetic Trees

- ▶ It can be very general: think of the tree of life

- ▶ It can be very general: think of the tree of life
- ▶ Nowadays, thanks to gene-sequencing techniques, we are building trees at the gene level (using sequence-alignment methods, i.e. comparing gene sequences to see how much similar/different two genes are between and within species (whattt!)).

- ▶ It can be very general: think of the tree of life
- ▶ Nowadays, thanks to gene-sequencing techniques, we are building trees at the gene level (using sequence-alignment methods, i.e. comparing gene sequences to see how much similar/different two genes are between and within species (whattt!)).
- ▶ A single phylogenetic tree can host multiple species

Annotated Phylogenetic Tree



Gene Functional Annotations: The Gene Ontology Project

Accession	GO:0060047
Name	heart contraction
Ontology	biological_process
Synonyms	heart beating, cardiac contraction, hemolymph circulation
Alternate	IDs None
Definition	The multicellular organismal process in which the heart decreases in volume in a characteristic way to propel blood through the body. Source: GOC:dph

Table 2 Heart Contraction Function. source: amigo.geneontology.org



pthr10037



pthr11521



pthr11521



pthr24356

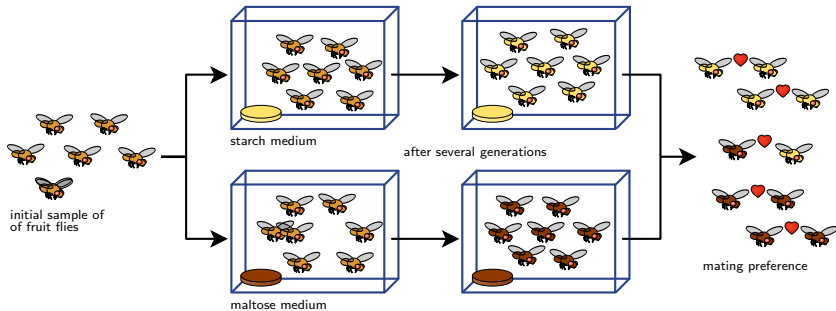


Figure 1 Dodd (1989): After one year of isolation, flies showed a significant level of assortativity in mating (wikimedia)

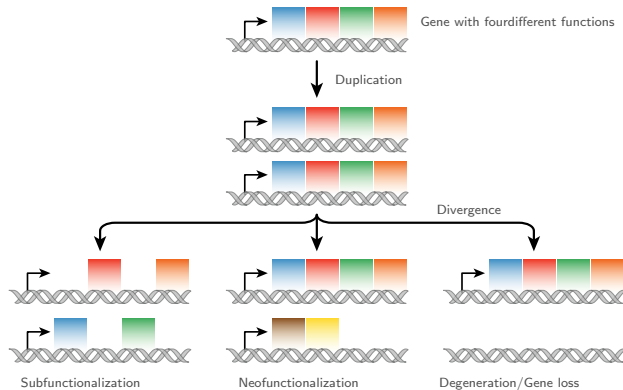


Figure 2 A key part of molecular innovation, gene duplication provides opportunity for new functions to emerge (wikimedia)

The general points of the model

- ▶ The rootnode in a phylogenetic tree is the best idea we have about the past, meaning, it could be that the tree has more behind, i.e. so functions may be gained since the beginning.

The general points of the model

- ▶ The rootnode in a phylogenetic tree is the best idea we have about the past, meaning, it could be that the tree has more behind, i.e. so functions may be gained since the beginning.
- ▶ At each step in evolution (interior node), there is a probability that the gene may gain/loss the function.

The general points of the model

- ▶ The rootnode in a phylogenetic tree is the best idea we have about the past, meaning, it could be that the tree has more behind, i.e. so functions may be gained since the beginning.
- ▶ At each step in evolution (interior node), there is a probability that the gene may gain/loss the function.
- ▶ Those probabilities vary depending on the type of the node: We believe that functional changes may happen at Duplication nodes.

An evolutionary model of gene functions: Formal statement

- ▶ The whole is based on the markov-assumption: The current state of the gene can be fully explained by its parent(s).

- ▶ The whole is based on the markov-assumption: The current state of the gene can be fully explained by its parent(s).
- ▶ For this we use Felsensteins' pruning algorithm (also known as...)

Formally

$$P(x = 1) = P(x = 1|x_p = 0)P(\text{Gain}) + P(x = 1|x_p = 1)P(\text{No loss})$$

The model has (so far) 7 fixed parameters

ψ_0, ψ_1 Probability of making a mistake (mislabel)

μ_{d0}, μ_{d1} Probability of functional gain/loss (duplication nodes)

μ_{s0}, μ_{s1} Probability of functional gain/loss (other nodes)

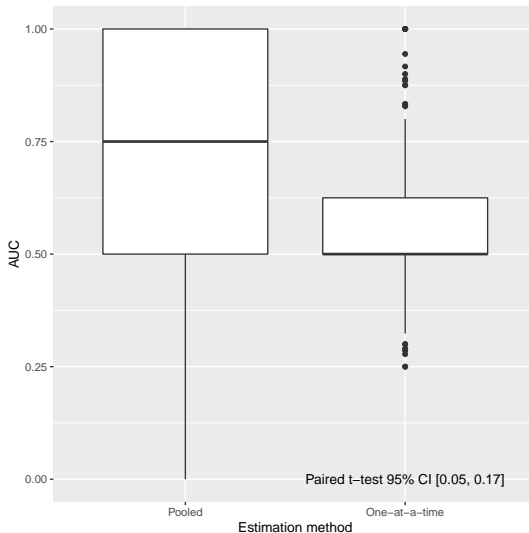
π Probability that the root has the function

- ▶ We developed a full set of tools (C++ library + R package) for this framework [▶ more](#)
- ▶ Estimation is done via: MLE, MAP, and MCMC (using an adaptive kernel)
- ▶ Posterior probabilities are estimated using the conditional on the observed data.
- ▶ To evaluate performance, we used two datasets: manually (fully) annotated (inferred) trees, and experimentally annotated trees

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
ψ_0	0.00	0.00	0.23	0.25	0.00	0.00	0.21	0.25
ψ_1	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01
μ_{d0}	0.01	0.01	0.97	0.96	1.00	0.01	1.00	0.98
μ_{d1}	0.01	0.02	0.52	0.58	0.25	0.02	0.51	0.58
μ_{s0}	0.00	0.00	0.05	0.06	0.07	0.00	0.05	0.06
μ_{s1}	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.02
π	0.81	0.91	0.78	0.45	0.82	0.91	0.83	0.49
Tree count	88	88	141	141	88	88	141	141
Method	MCMC	MCMC	MCMC	MCMC	MLE	MLE	MLE	MLE
Prior	Uniform	Beta	Uniform	Beta	Uniform	Beta	Uniform	Beta
Inferred	Yes	Yes	No	No	Yes	Yes	No	No
AUC	1.00	1.00	0.69	0.67	0.98	1.00	0.70	0.67
P. Score (obs)	1.00	1.00	0.81	0.81	0.92	1.00	0.81	0.81
P. Score (random)	0.71	0.71	0.61	0.61	0.71	0.71	0.61	0.61

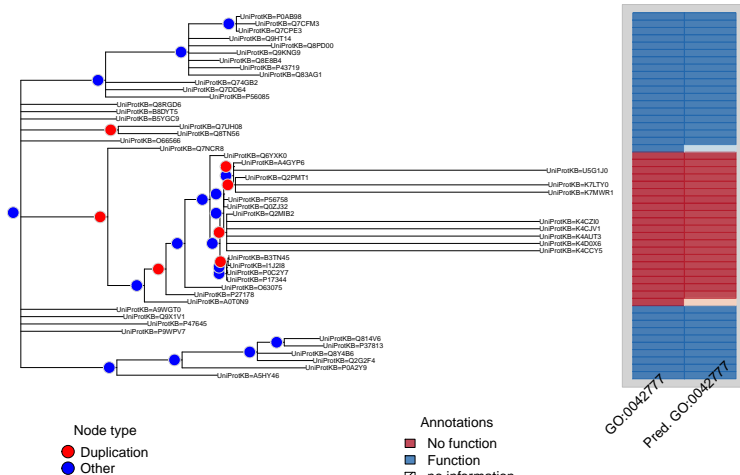
Table 3 Parameter estimates using different estimation methods, priors, and types of annotations.

Paper 2: Pooled estimation (worth it?)



Paper 2: Leave-one-out predictions

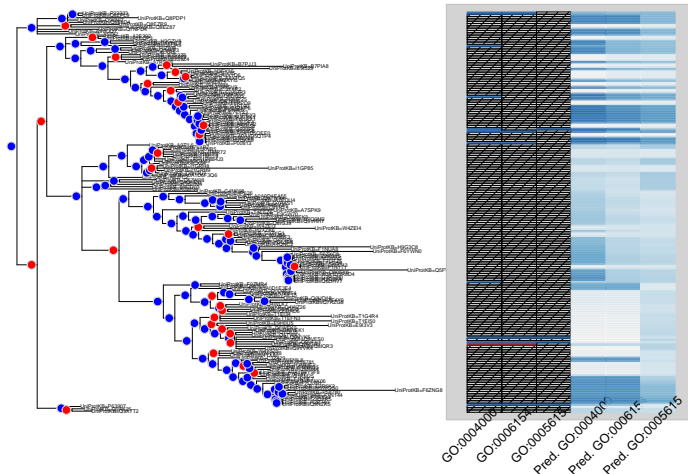
Annotated Phylogenetic Tree



Paper 2: Out-of-sample-predictions

Adenosine Deaminase (PTHR11409)

AUCs:={0.80, 0.67, -}



Paper 2: Key takeaways

- ▶ (Yet another) model for predicting gene functions using phylogenetics

- ▶ (Yet another) model for predicting gene functions using phylogenetics
- ▶ Big difference... computationally scalable

- ▶ (Yet another) model for predicting gene functions using phylogenetics
- ▶ Big difference... computationally scalable
- ▶ Meaningful biological results

- ▶ (Yet another) model for predicting gene functions using phylogenetics
- ▶ Big difference... computationally scalable
- ▶ Meaningful biological results
- ▶ Preliminary accuracy results comparable to state-of-the-art phylo-based models

Possible venues to continue

Possible venues to continue

- Identify an adequate test for goodness-of-fit assesment

Possible venues to continue

- ▶ Identify an adequate test for goodness-of-fit assesment
- ▶ Extend to estimation of large graphs by splitting the networks in induced-subgraphs

Future directions: Gene functional prediction

Possible venues to continue

Possible venues to continue

- ▶ Incorporate more external information using leaf(and node?) level features.

Possible venues to continue

- ▶ Incorporate more external information using leaf(and node?) level features.
- ▶ Adapt the model to incorporate joint estimation of functions using pseudo-likelihood.

$$P(a, b, c) \approx P(a, b)P(b, c)P(a, c)$$

Possible venues to continue

- ▶ Incorporate more external information using leaf(and node?) level features.
- ▶ Adapt the model to incorporate joint estimation of functions using pseudo-likelihood.

$$P(a, b, c) \approx P(a, b)P(b, c)P(a, c)$$

- ▶ Make the model hierarchical when pooling trees: different mutation rates.

Here are some by-products of my research here at USC

- ▶ The slurmR R package
- ▶ The pruner C++ library
- ▶ The fmcmc R package

Dodd, D. M. B. (1989). Reproductive isolation as a consequence of adaptive divergence in *Drosophila pseudoobscura*. *Evolution*, 43(6), 1308–1311. Retrieved from <http://www.jstor.org/stable/2409365>

One of the most popular methods for estimating ERGMs is the MC-MLE approach (citations here)

This consists on the following steps

1. Start from a sensible guess on what should be the population parameters (usually done using pseudo-MLE estimation)
2. While the algorithm doesn't converge, do:
 - 2.1 Simulate a stream of networks with the current state of the parameter, θ_t
 - 2.2 Using the law of large numbers, approximate the ratio of likelihoods based on the parameter θ_t , this is the objective function
 - 2.3 Update the parameter by a Newton-Raphson step
 - 2.4 Next iteration

- ▶ Implements estimation of ERGMs using exact statistics for small networks
- ▶ Metaprogramming allows specifying likelihood (and gradient) functions for joint models
- ▶ Includes tools for simulating, and postestimation checks
- ▶ Getting ready for CRAN!

◀ go back

► TBF

◀ go back