# Analysis Assignment #3: Random Graph Models Instructions

### Final submission to be uploaded to Laulima Saturday February 19th.

### 50 points total possible

These are the instructions, which you should read carefully before starting.

Then use Analysis-3-Random-Models-Lastname.Rmd as template to construct your report.

## Prelude

A Graph Model is a probability distribution over a collection or ensemble of possible graphs, constrained by a set of parameters. There are a variety of different graph models: this week we studied random graphs (constrained only by number of vertices and either number or probability of edges), configuration model graphs (also constrained by degree sequence), degree preserving randomization of a given graph, and the Watts-Strogatz small world model. Graph models can be useful in Network Science research as 'points' of comparison: one might choose a graph model that one thinks matches the natural network under study in some aspects (such as degree distribution or average shortest path length), randomizing other aspects, and then see in what ways the natural network differ from the graph model. The departures are properties that we still need to explain.

In the main part of this analysis assignment, you will try to match 3 of the graph models we have studied to natural networks. You will adjust the parameters of the graph models to get them to fit as closely as possible, and then run our usual suite of metrics (from Week 4 and Analysis 2) to see where each model matches the natural network and where each departs or fails to model the property measured by the metric. The objectives of this exercise are (a) to help you understand the properties of the models and (b) to become comfortable with the process of fitting a model to a graph. We'll also learn some more R in the process.

### R Background

We will use R data frames to compile the results. Unless you are already familiar with matrices and data frames, run swirl R tutorial #7 on matrices and data frames. (You should have already done the previous tutorials, especially #4 on vectors.)

* library(swirl)
* swirl()
* choose 1: R Programming
* choose 7: Matrices and Data Frames

Then review 5-4-Data-Frames.R, the demo I showed in class, and the associated video if you wish, for three ways to use data frames or tibbles: choose the approach you prefer. That class demo was on the Network Science collaboration graph. For the homework you will do something nearly identical on two other graphs.

### Networks

We will use these networks, which may be found in the Analysis 3 folder. (You can find documentation in their subfolders of the Networks folder.) The .Rmd template is already set up to load them.

* EuroSiS WebAtlas (WA): EuroSiS-WebAtlas-Simplified.graphml, the *simplified* version we created with N=1285 nodes and L=6462 links.
* Internet Routers (IR): internet\_routers-22july06.graphmlwith N=22963 and L=48436.

## Problems

### 1. Prediction based on Random Graphs (10 pts)

Here we predict the connectivity of EuroSiS WebAtlas and Internet Routers based on the assumption that they are random graphs. If they depart from the prediction this tells us something about how they are not random.

**(a)** Compute the average degree <k> and the natural log ln(|V|) for each of these networks, assigning the results to variables given. Then uncomment the references to these variables in the table so they display.

**(b)** Using the above parameters and Barabasi Figure 3.7, predict

* the regime a random model G(n,m) of each of the above networks will be in, and
* specifically whether it will have many small components, a mixture of small and a giant component, or be fully connected.

Write a few sentences explaining how the prediction follows from <k> and ln(|V|).

**(c)** Construct a table of component size distribution to find out how many components they actually have. Do this in the .Rmd template to display the table of component sizes.

**(d)** Explain why the predictions do or do not match the actual number of components. This is graded on whether you show that you can reason plausibly about random graphs as models for real world phenomena, so you should talk about the properties or processes of the real world phenomena. This need not be long: write a brief paragraph of discussion for each network.

### 2. Analysis (30 pts, 15 per network)

**In R/igraph,** model the two networks EuroSiS WebAtlas and Internet Routers. (Computing distances in Internet Routers can be slow. Start early, or do just EuroSiS for half the points.) Model them with the following models:

* **G(n,m) Random Graphs**: sample\_gnm
* **Configuration model**, constrained by degree sequence of your network: sample\_degseq
* **Degree Preserving Randomization**: rewire with keeping\_degseq, and a factor of ecount \* 1000 (this will be slow with Internet Routers: you may use 100)

(Comment: in previous years this assignment included the Watts-Strogatz Small-World model, but I decided to remove it because it is not commonly used and can be more difficult to match the edge count.)

Evaluate each model by running these metrics on both the model graphs and the natural network graph and comparing:

* **|V|**: vcount(g)
* **|E|**: ecount(g)
* **mean geodesic distance**: mean\_distance(g)
* **global transitivity**: transitivity(g, type="global")
* **component distribution**: count\_components(g)
* **degree assortativity**: assortativity\_degree(g)

Collect the results in a data frame, using your choice of method from 5-4-Data-Frames.R, so that you can display a table of results in the .Rmd document simply by printing the data frame. I highly recommend the tibble method!

Also, plot degree distributions of the natural network and of G(n,m). (You don't need to plot the degree distribution of the configuration model and degree preserving randomization, as they are guaranteed to be identical to the natural model.)

**Your .Rmd document will include the following items** and the code to generate them:

* **(a) Summary: summary(g) of all the networks g.**
* **(b) Results of the metrics for all networks** **as collected in the data frame**.
* **(c) Degree distribution plots** for the natural network and G(n,m).
* **(d) Discussion of how each model matches or fails to match the natural network.** What is each type of graph model good at modeling? What might be operating in the natural network that each model does not capture? ***This part is important: don't skimp on your thinking.*** You already discussed component distribution in 1(d): focus here on processes captured or not captured by the other metrics. Be sure that your discussion is not generic and vague, but rather is based on *domain processes specific to the network being modeled.*

### 3. Visualization and Interpretation - 10 pts

**Do this only for EuroSiS WebAtlas**

**(a)** Write out *all 3* of the model networks as graphml files (G(n,m), configuration, and degree preserving randomization).

**(b)** Read the original graph and all 3 models into Gephi.

**(c)** Run the *same* visualization algorithm/procedure on all 4 graphs so that you can compare them. First set node size to be proportional to degree, so we can see the effect of the degree distributions. Then choose a visualization algorithm and parameter settings such that you can run it from random layout and reach equilibrium with a good visualization. You will need to find a visualization that shows the structure in all 4 networks. (Do not turn on labels. There is no need to color the nodes.)

Capture bitmap images using the Camera tool in Overview (bitmap is sufficient, as we only need to see the overall structure of each visualization), and embed the images in the .Rmd document.

Also save the .gephi project and include that in your uploaded solution folder, in case we need to look at it.

**(d)** Discuss how the visualizations show the difference between the graph models. This will include some of the differences you noted in 2c, but now you are discussing them as manifest in the visualization rather than in the metric numbers. So, your discussion should reference what you see in the visualization.