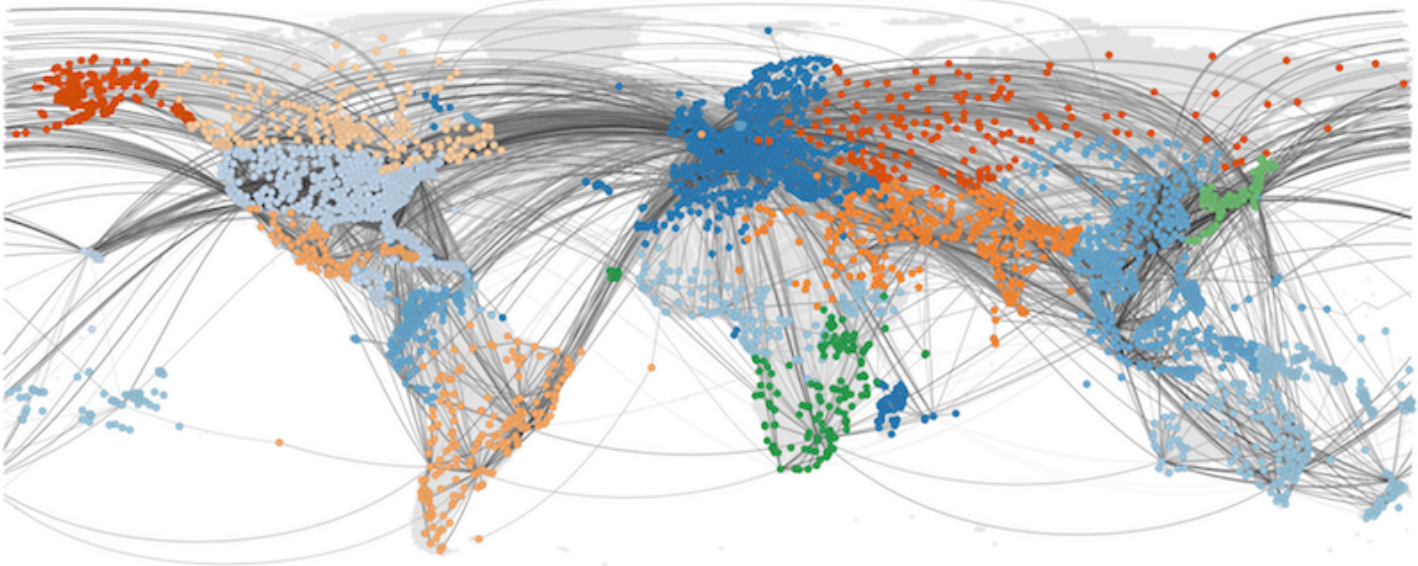
Large network structure and random graph models

You are given a selection of **real networks** of **various size and origin**.

- Zachary's karate club network (34 nodes)
- Davis's southern women network (32 nodes)
- Lusseau's bottlenose dolphins network (62 nodes)
- Ingredients network by common compounds (1,525 nodes)
- Map of Darknet from Tor network (7,178 nodes)
- Human protein-protein interaction network (19,634 nodes)
- Internet map of autonomous systems (75,885 nodes)
- Amazon product copurchase network (262,111 nodes)
- Paper citation network of APS (438,943 nodes)
- Small part of Google web graph (875,713 nodes)
- Road/highway network of Texas (1,379,917 nodes)

All networks are available in Pajek, edge list and LNA formats.



I. Toy network construction and Pajek format

1. Using your programming library, **construct a small toy network** with a few nodes and edges. **Print out** its name, and the **number of nodes and edges**.

2. Using the methods provided by your programming library, **read in real networks** in Pajek format

above and **print out their basic statistics**. What is the size of the largest network you are able to construct in say a minute?

II. Basic network statistics, connectivity, distances etc.

1. Compute and print out **basic statistics of real networks** above. Namely, the number of nodes n, the number of isolated nodes n_0 , the number of edges m, the number of self-edges or loops m_0 , the average node degree $\langle k \rangle = 2m/n$ and the undirected density $\rho = \langle k \rangle/(n-1)$. Are the results **expected or surprising**? *(Computational complexity of these computations is either constant $\mathcal{O}(1)$ or linear $\mathcal{O}(m)$ and should be applicable to any network you are able to fit in your memory.)*

2. Using **depth-first search** methods provided by your programming library, compute **connected components of real networks** above. Print out the fraction of nodes in the largest connected component S and the number of all connected components s. Are the results **expected or surprising**? *(Computational complexity of these computations is linear $\mathcal{O}(m)$ and should be applicable to any network you are able to fit in your memory.)*

3. Using **breadth-first search** methods provided by your programming library, compute **distances between the nodes of real networks** above. Print out the average distance between the nodes $\langle d \rangle$ and the maximum distance or diameter d_{max} . Are the results **expected or surprising**? *(Computational complexity of these computations is inevitably quadratic $\mathcal{O}(nm)$ and you should use approximation implemented in the started script.)*

4. Using **triad counting** methods provided by your programming library, compute **node clustering coefficient of real networks** above. Print out the average node clustering coefficient $\langle C \rangle$. Are the results **expected or surprising**? *(Computational complexity of these computations is superlinear $\mathcal{O}(m\langle k \rangle)$ and should be applicable to all but the largest networks.)*

5. *(tentative)* Using **plotting functionality** provided by your programming library, compute **degree distribution of real networks** above. Plot degree distribution p_k on a doubly logarithmic plot. Are the results **expected or surprising**?*(Computational complexity of these computations is linear $\mathcal{O}(n)$ and should be applicable to any network you are able to fit in your memory.)*

6. What is the **size of the largest network** you are able to analyze in say a minute?

III. Random graphs, scale-free and small-world network models

1. Construct **Erdös-Rényi random graphs** $G(n, m)$ with the same number of nodes n and edges m as **real networks** above. Print out their standard statistics. Are the results **expected or surprising**?

2. *(tentative)* Next, construct **Barabási-Albert scale-free networks** $G(n, \langle k \rangle/2)$ with the same number of nodes n and the average degree $\langle k \rangle$ as **real networks** above. Print out their standard statistics. Are the results **expected or surprising**?

3. *(tentative)* Finally, construct also **Watts-Strogatz small-world networks** $G(n, \langle k \rangle, p)$ with the same number of nodes n and edges m as **real networks** above. Print out their standard statistics. Are the results **expected or surprising**?