Network Science

Communities Part 1

Sean P. Cornelius
With
Emma K. Towlson and Albert-László Barabási

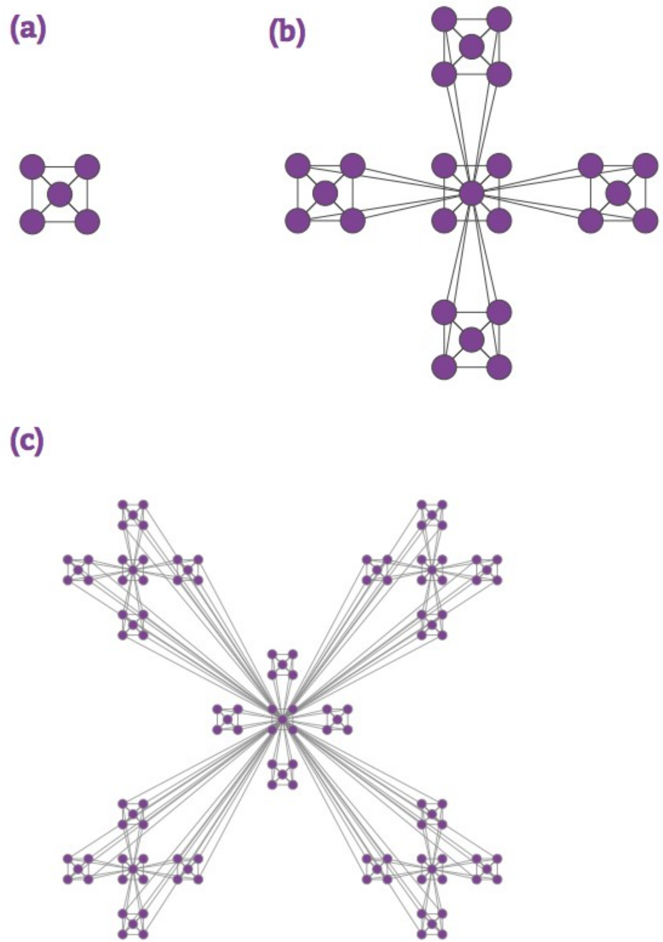
www.BarabasiLab.com

i. Nested Communities

It assumes that communities are organized in a hierarchical fashion, i.e. small modules are nested into larger ones. This hierarchical nesting is captured by the dendrogram (Figures 9.12a and 9.15e). How do we know, however, if such hierarchy is indeed present in a network? Could this hierarchy be imposed by the algorithm, whether or not the underlying network has a nested community structure?

ii. Communities and the Scale-Free Property

The density hypothesis states that a network can be partitioned into a collection of subgraphs that are only weakly linked to other subgraphs. How can we have somewhat isolated communities in a scale-free network, whose hubs inevitably connect to nodes that can belong to different communities?

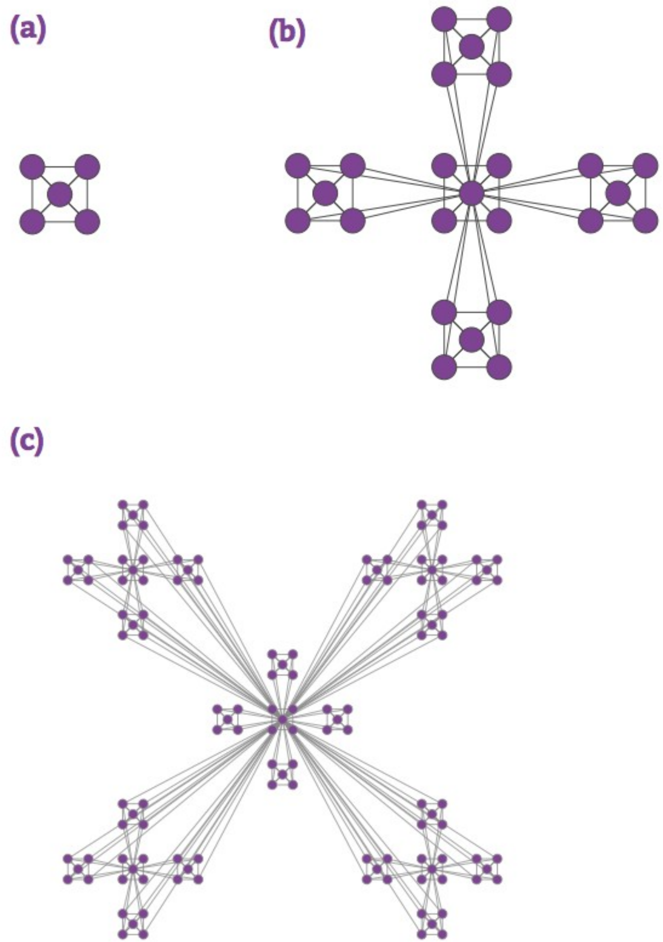


(1) Scale-free property

The obtained network is scale-free, its degree distribution following a power-law with $$\gamma = 1 + \frac{\ln 5}{\ln 4} \simeq 2.16$$

E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).
A.-L. Barabási, *Network Science: Communities*.

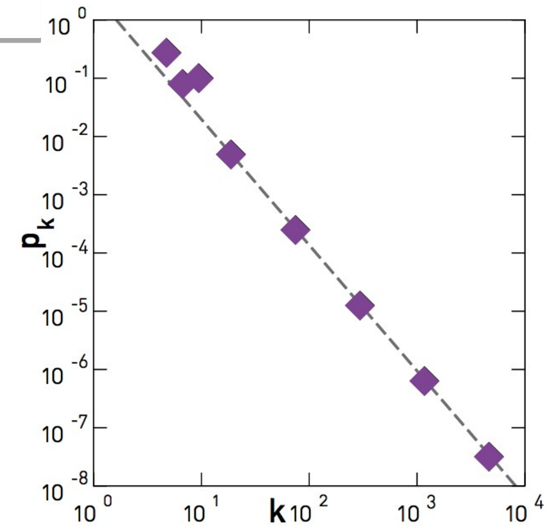
Section 4 Hierarchy in networks



(1) Scale-free property

The obtained network is scale-free, its degree distribution following a power-law with

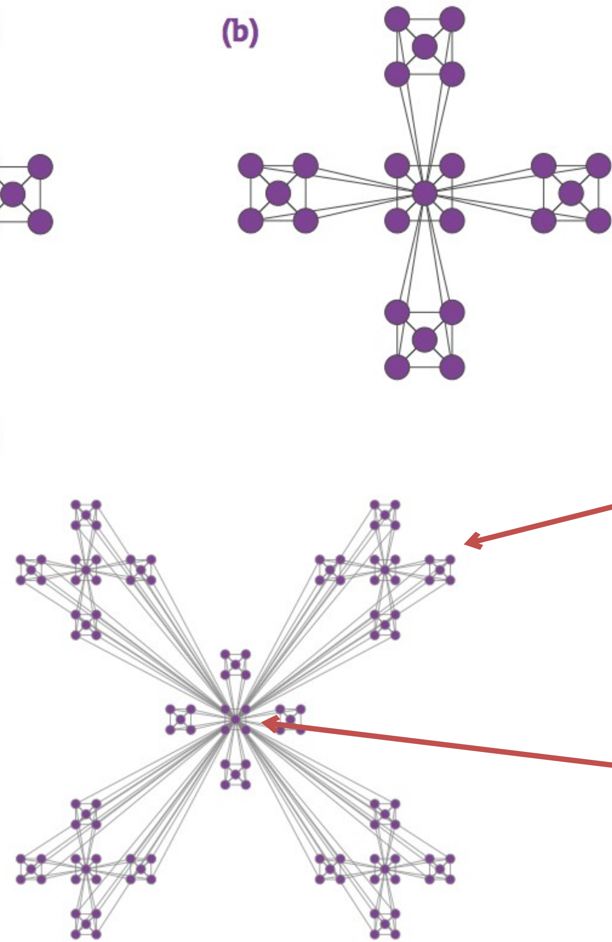
$$\gamma = 1 + \frac{\ln 5}{\ln 4} \simeq 2.16$$



E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities*.

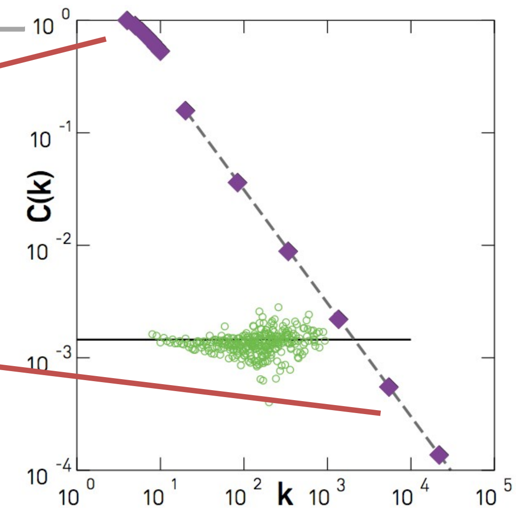
(a) (b) (c)



(2) Clustering coefficient scaling with k

$$C\left(k\right)=\frac{\#\ \text{between}\ k\ \text{neighbors}}{k\left(k-1\right)/2}=$$

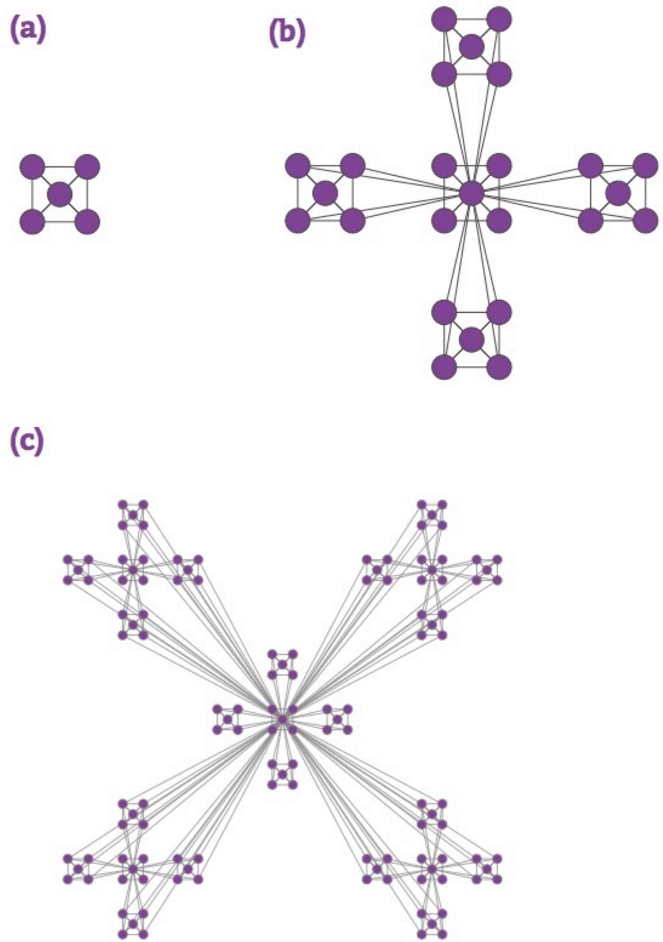
$$C\left(k\right)\sim k^{-1}$$



<u>Small *k* nodes:</u>
*high clustering coefficient;
*their neighbors tend to link to each other in highly interlinked, compact communities.

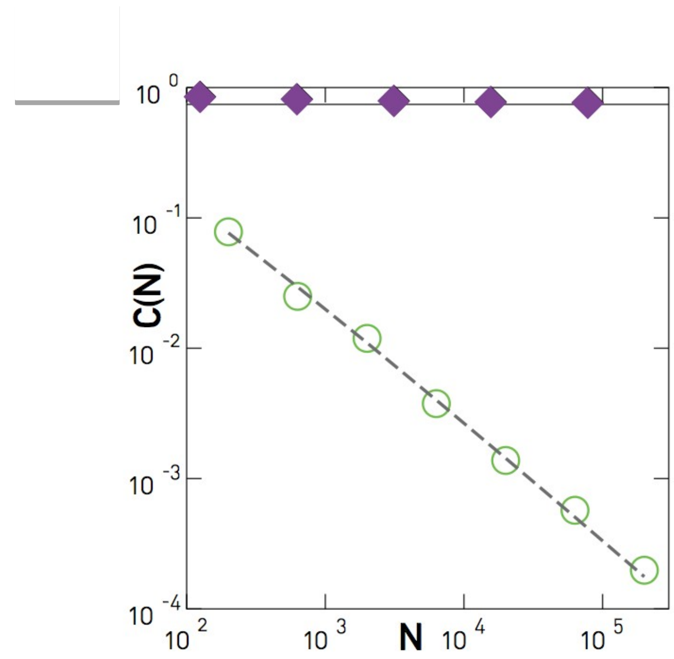
<u>High *k* nodes (hubs):</u>
*small clustering coefficient;
*connect independent communities.

E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).
A.-L. Barabási, *Network Science: Communities*.



(3) Clustering coefficient independent of N

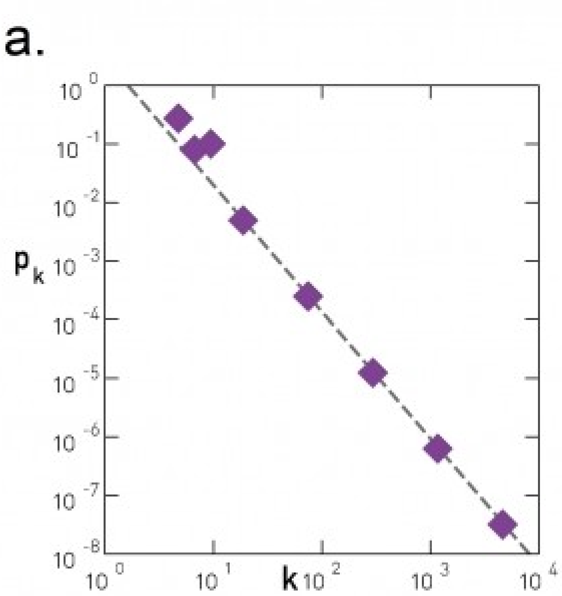
$$C = 0.743$$



E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).
A.-L. Barabási, *Network Science: Communities*.

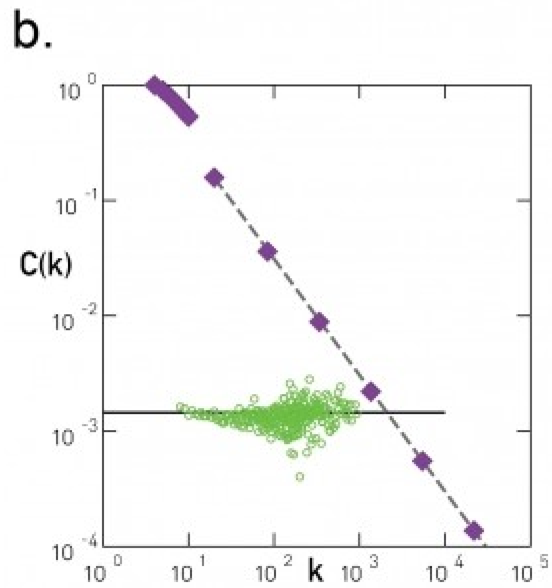
1. Scale-free

$$\gamma = 1 + \frac{\ln 5}{\ln 4} = 2.161$$



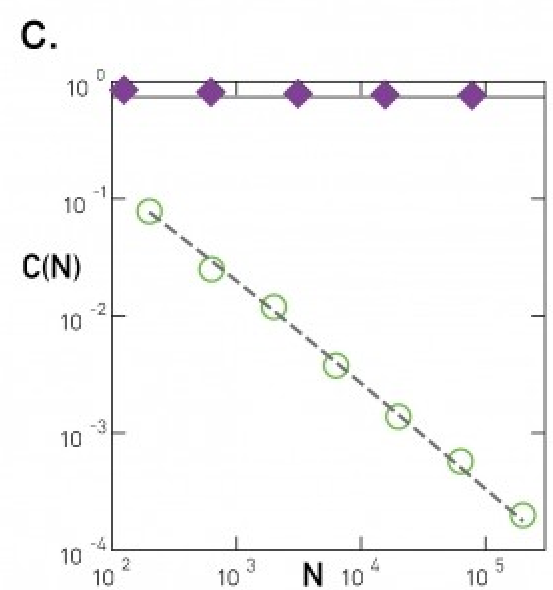
2. Scaling clustering coefficient (DGM)

$$C(k) \sim k^{-1}$$



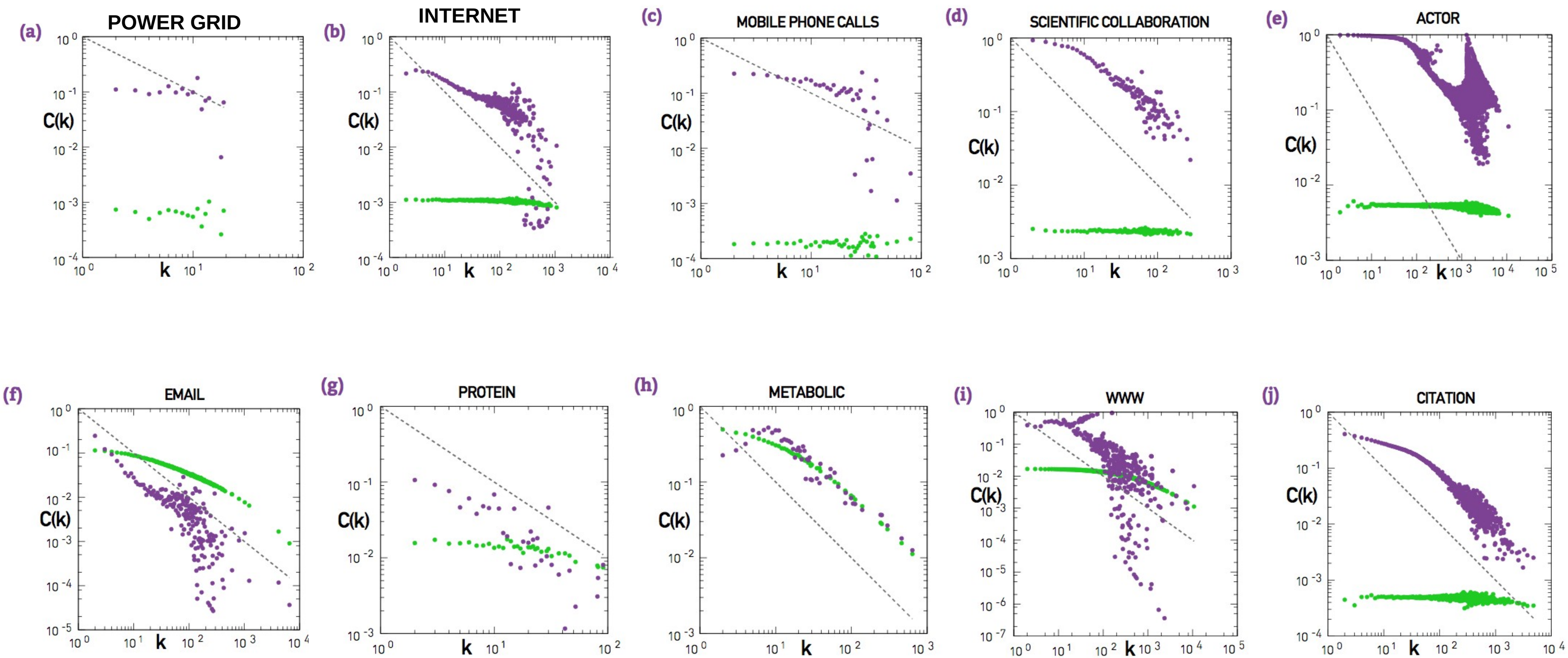
3. Clustering coefficient independent of N

$$C(N) = \text{const.}$$

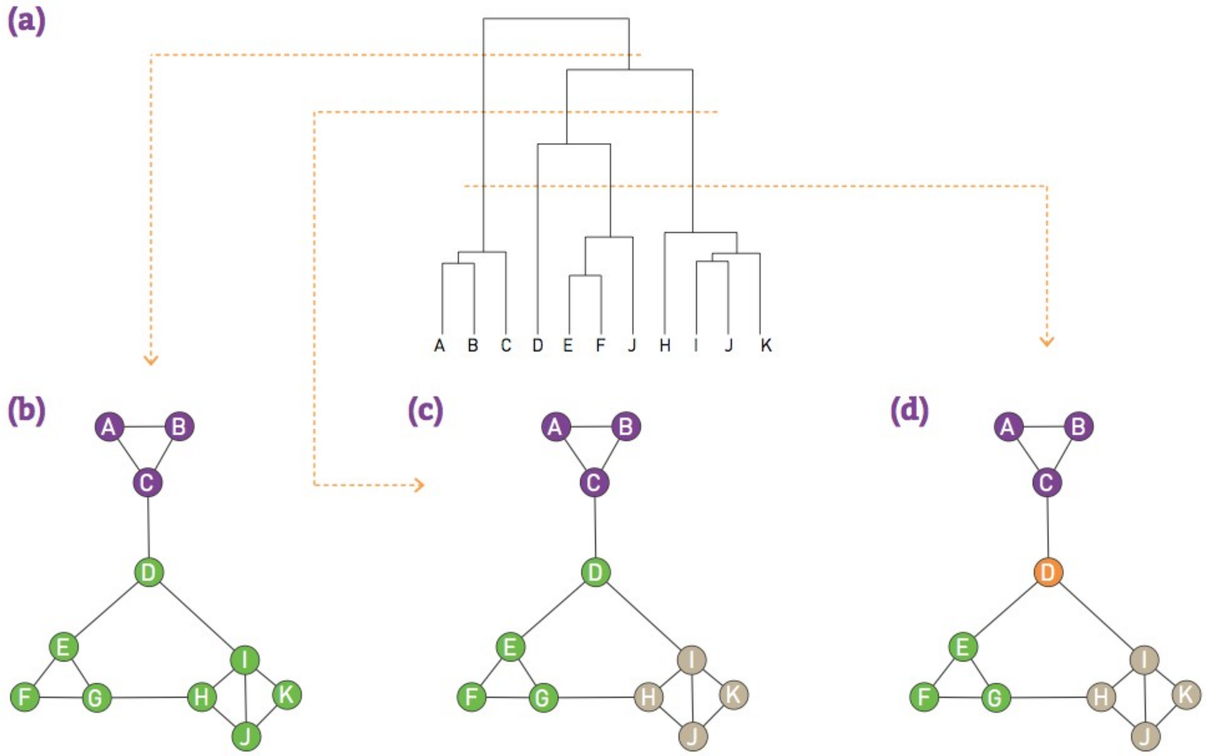


E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).
A.-L. Barabási, *Network Science: Communities*.

Section 4 Hierarchy in real networks



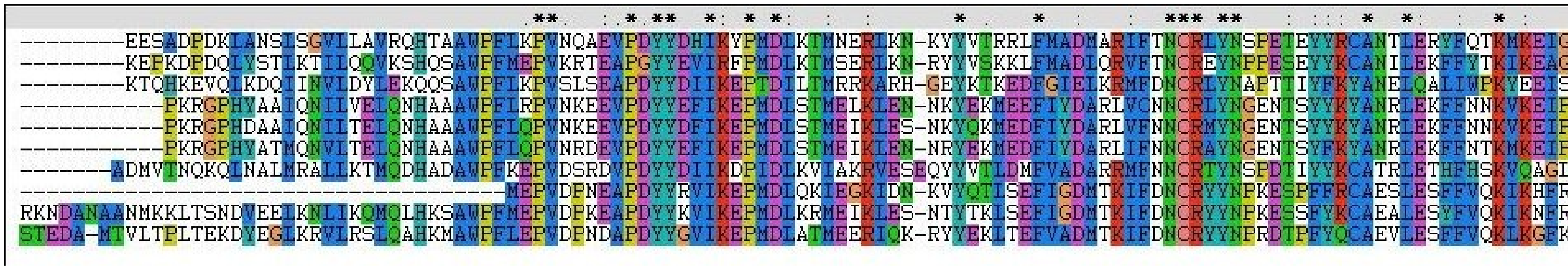
Where to "cut"?



A.-L. Barabási, *Network Science: Communities*.

Phylogenetic dendrograms

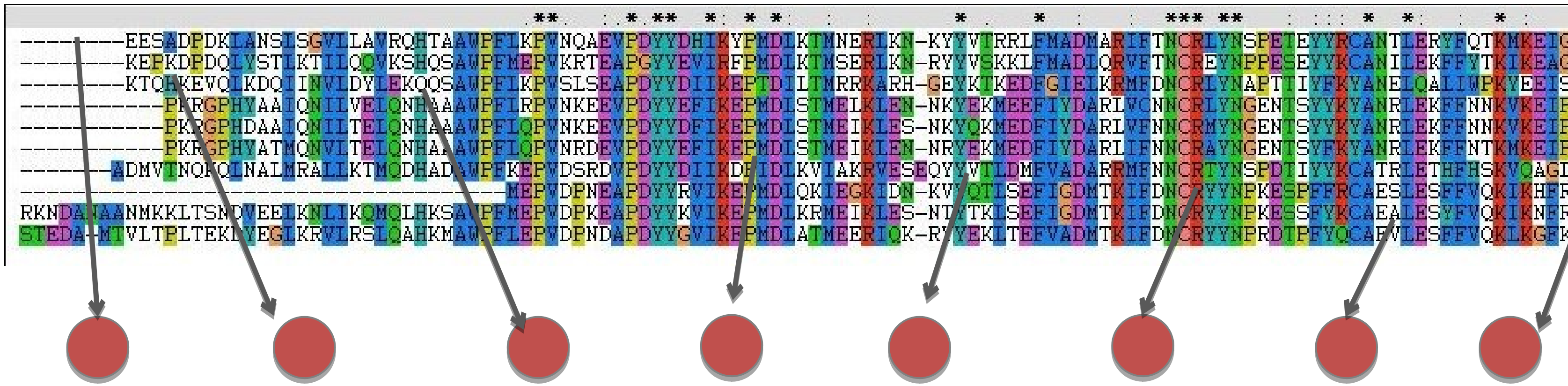
In bioinformatics, clusters and dendrograms have been studied for a long time.



For example, the sequences of the same protein or gene in different species are selected, and compared with each other.

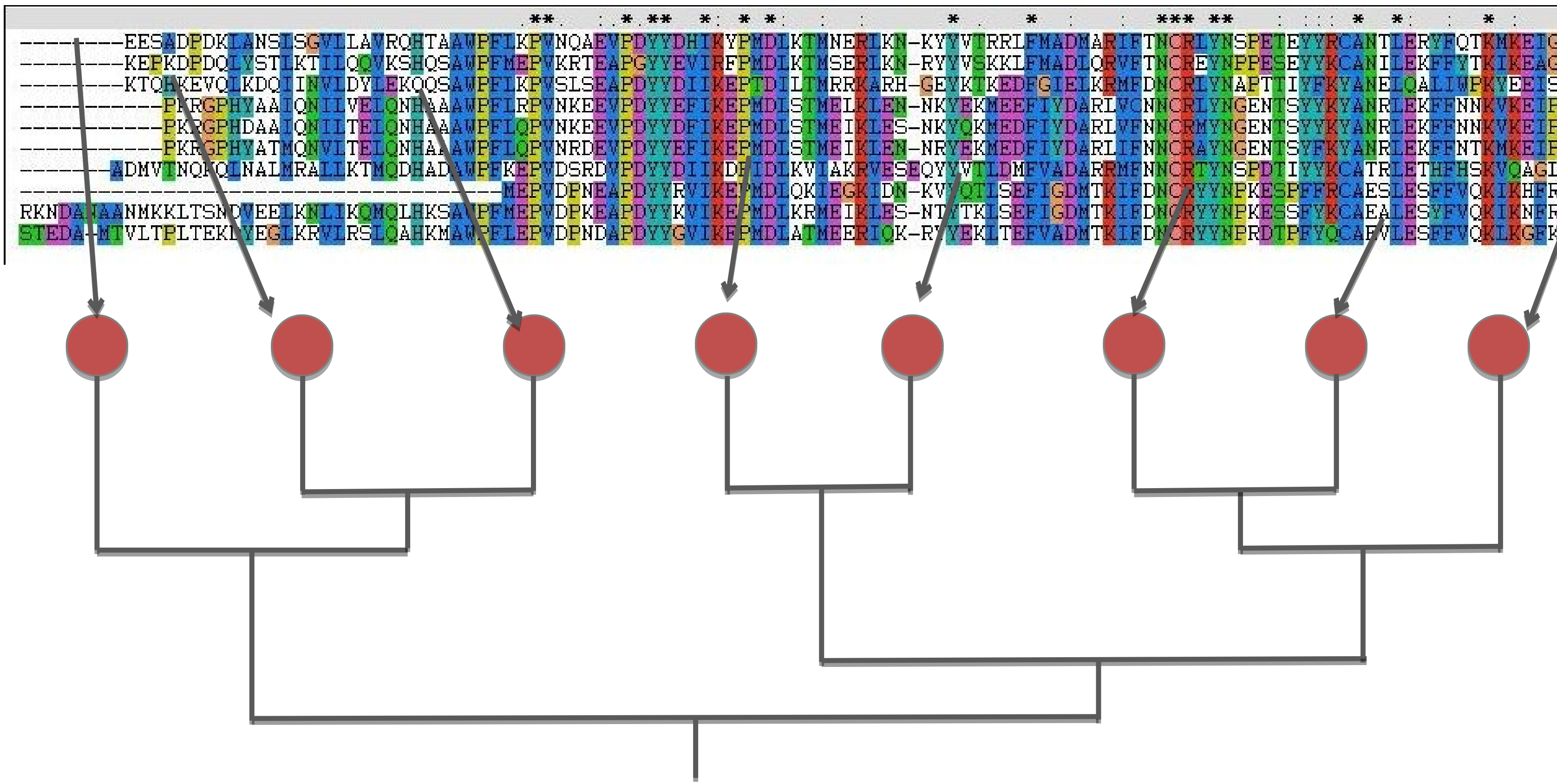
Phylogenetic dendrograms

A similarity matrix is constructed between these sequences, by looking at how many aminoacids/nucleotides stay in place

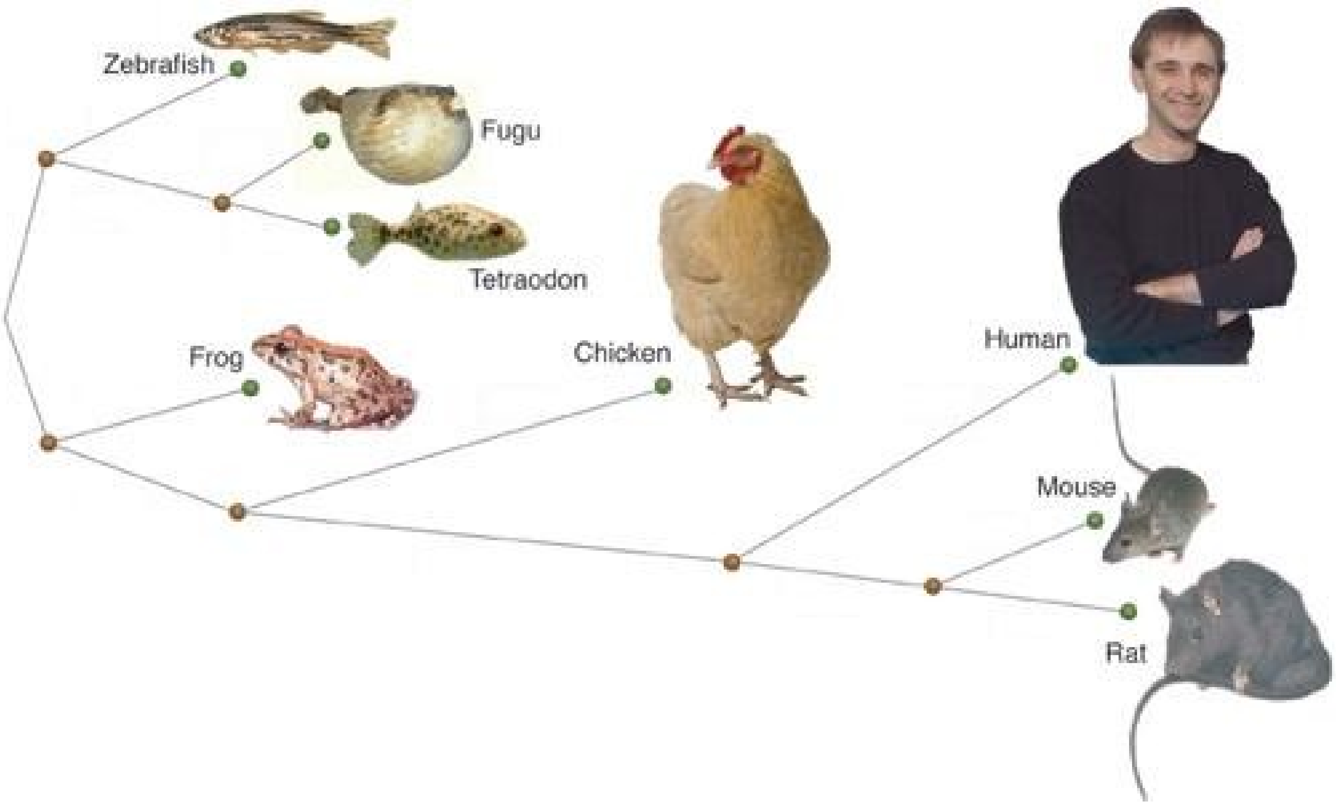


Phylogenetic dendrograms

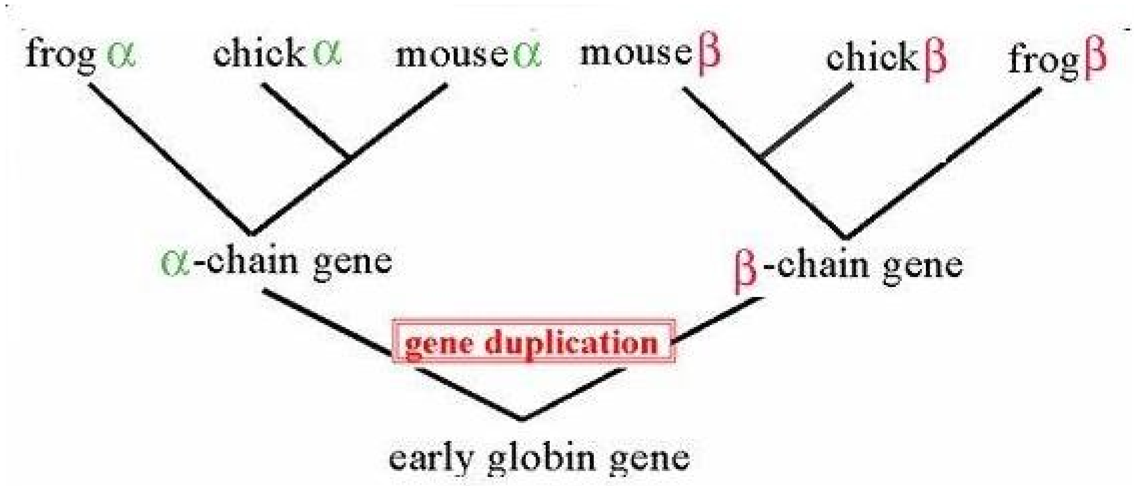
A similarity matrix is constructed between these sequences, by looking at how many aminoacids/nucleotides stay in place



Phylogenetic dendrograms



Phylogenetic dendrograms



Modularity

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

MEJ Newman, *PNAS* 103 (2006).
A.-L. Barabási, *Network Science: Communities*.

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M(C_c) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - p_{ij})\,\delta(C_i - C_j)$$

MEJ Newman, *PNAS* 103 (2006).

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M(C_c) = \frac{1}{2L} \sum_{i,j=1}^{N} (A_{ij} - p_{ij})\, \delta(C_i - C_j)$$

Original data

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $M(C_c) = \frac{1}{2L} \sum_{i,j=1}^{N} (A_{ij} - p_{ij})\, \delta(C_i - C_j)$

Original data

Expected connections in a random model

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity
$$M(C_c) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - p_{ij})\,\delta(C_i - C_j)$$

Original data

Expected connections in a random model

Relative to a specific partition

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M(C_c)=\frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij}-p_{ij})\delta(C_i-C_j)$$

Original data

Expected connections in a random model

Relative to a specific partition

→ Random network $p_{ij}=2L\,p_i\,p_j=\frac{k_i k_j}{2L}$

→ Modularity is a measure associated to a partition

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

Another way of writing *M*

$$M(C_c)=\frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij}-p_{ij})\delta(C_i-C_j) \qquad p_{ij}=2L\,p_i\,p_j=\frac{k_i k_j}{2L}$$

We can rewrite the first term as

$$\frac{1}{2L}\sum_{i,j=1}^{N}A_{ij}\delta(C_i-C_j)=\sum_{c=1}^{n_c}\frac{1}{2L}\sum_{i,j\in C_c}A_{ij}=\sum_{c=1}^{n_c}\frac{l_c}{L}$$

where l_C is the number of links within C. In a similar fashion, the second term becomes

$$\frac{1}{2L}\sum_{i,j}\frac{k_i k_j}{2L}\delta(C_i-C_j)=\sum_{c=1}^{n_c}\frac{1}{(2L)^2}\sum_{i,j\in C_c}k_i k_j=\sum_{c=1}^{n_c}\frac{k_c^2}{4L^2}$$

Finally we get:

$$M(C_c)=\sum_{c=1}^{n_c}\left[\frac{l_c}{L}-\left(\frac{k_c}{2L}\right)^2\right]$$

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

H5: ***Maximal Modularity Hypothesis***

The partition with the maximum modularity M for a given network offers the optimal community structure

MEJ Newman, *PNAS* 103 (2006).
A.-L. Barabási, *Network Science: Communities*.

H5: *Maximal Modularity Hypothesis*

The partition with the maximum modularity *M* for a given network offers the optimal community structure

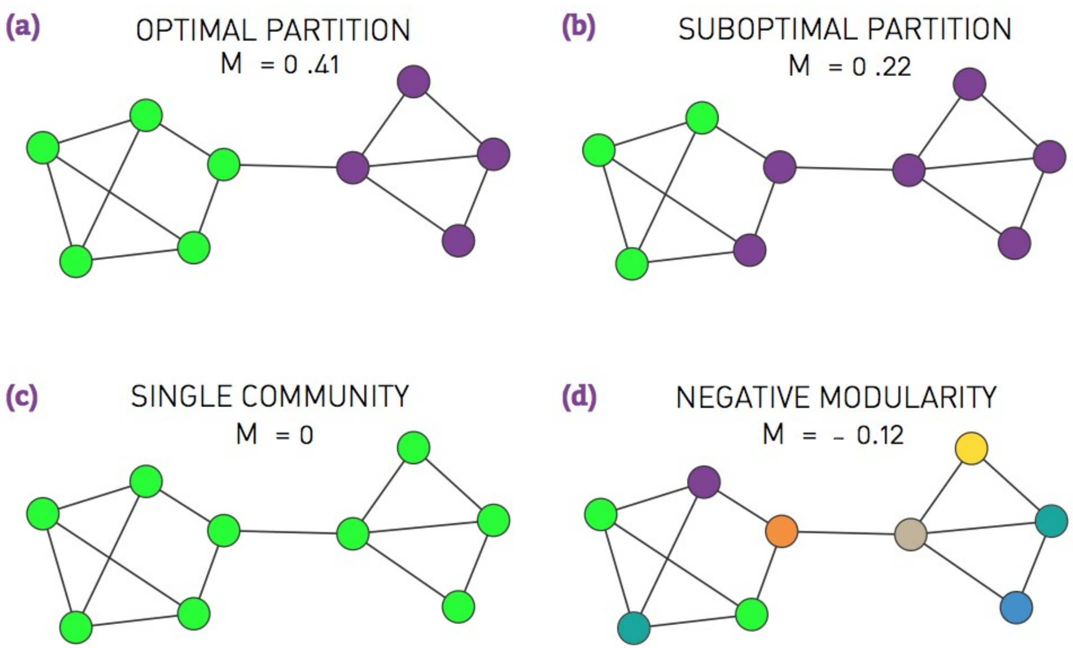
Goal

Find $\{C_c, c = 1, n_c\}$ that maximizes M

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities*.

Which partition $\{C_c,\ c = 1,\ n_c\}$?



- *Optimal partition*, that maximizes the modularity.
- *Sub-optimal* but positive modularity.
- *Negative Modularity*: If we assign each node to a different community.
- *Zero modularity:* Assigning all nodes to the same community, independent of the network structure.
- *<u>Modularity is size dependent</u>*

Modularity based community identification

A *greedy algorithm*, which iteratively joins nodes if the move increases the new partition's modularity.

Step 1. Assign each node to a community of its own. Hence we start with N communities.

Step 2. Inspect each pair of communities connected by at least one link and compute the modularity variation obtained if we merge these two communities.

Step 3. Identify the community pairs for which ΔM is the largest and merge them. Note that modularity of a particular partition is always calculated from the full topology of the network.

Step 4. Repeat step 2 until all nodes are merged into a single community.

Step 5. Record for each step and select the partition for which the modularity is maximal.

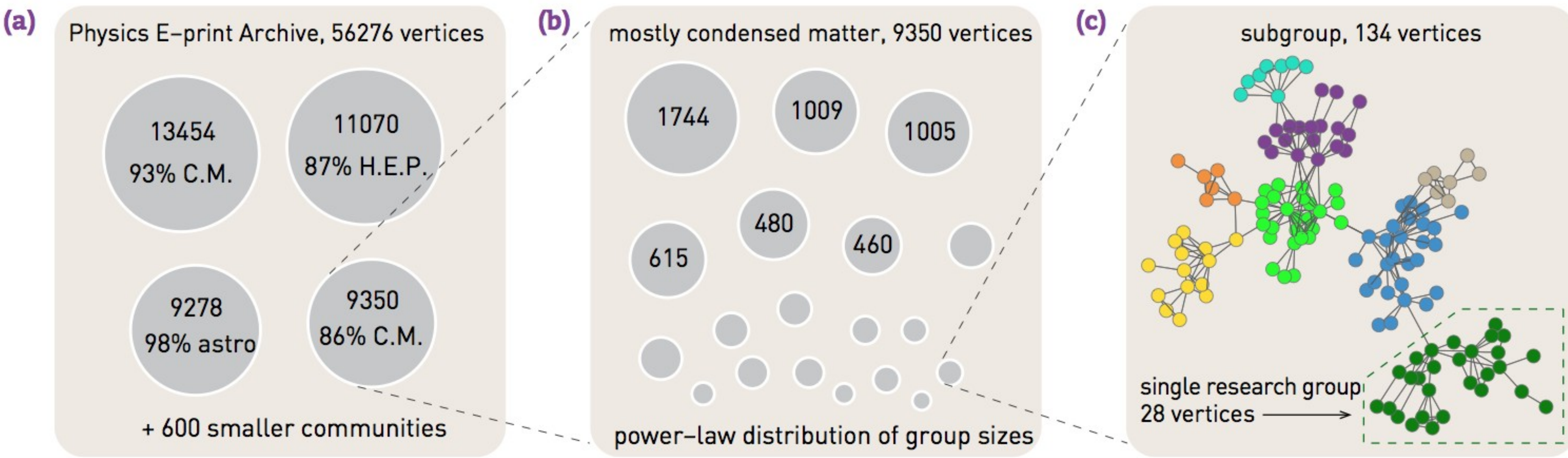
MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities*.

Which partition $\{C_c,\ c = 1,\ n_c\}$ **?**

→ It can be used to design new algorithms, **aiming at maximizing *M***

→ Modularity can be used to compare different partitions provided by other algorithms, like hierarchical clustering



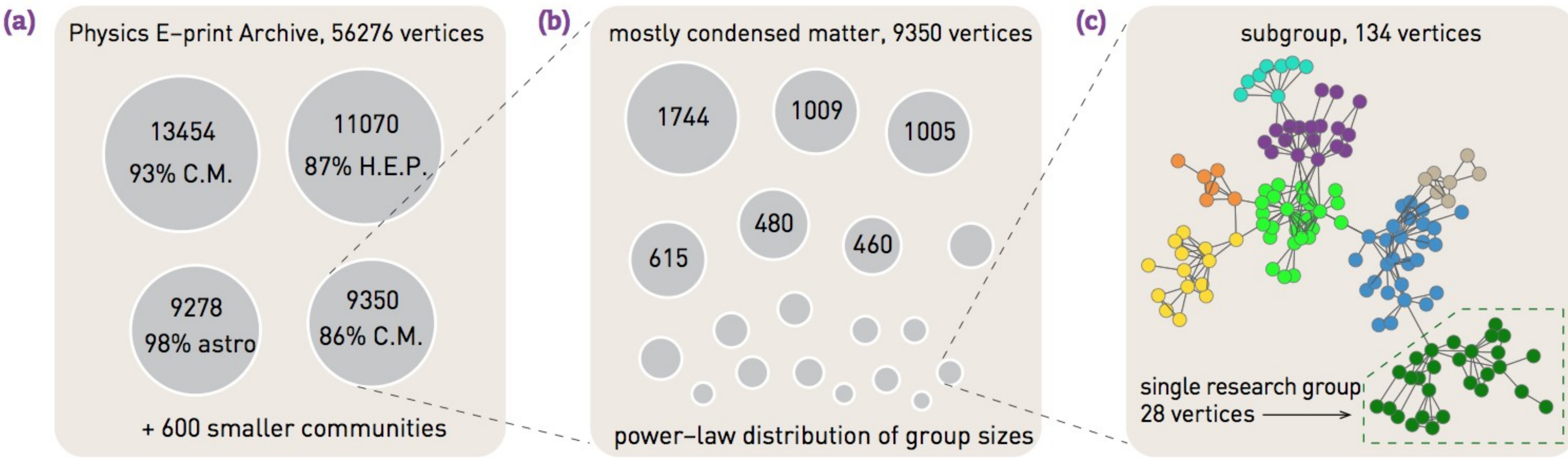
Computational complexity:

- Step 1-2 (calculation of ΔM for L links): $O(L)$
- Step 3 (matrix update): $O(N)$
- Step 4 (N-1 community merges): $O((L+N)N)$

for sparse networks

MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities*.



Computational complexity:

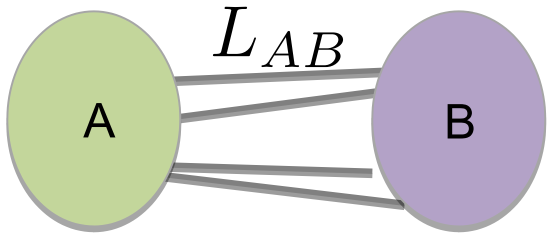
- Step 1-2 (calculation of ΔM for L links): $O(L)$
- Step 3 (matrix update): $O(N)$
- Step 4 (N-1 community merges): $O((L+N)N)$

for sparse networks → $O(N^2)$

MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities*.

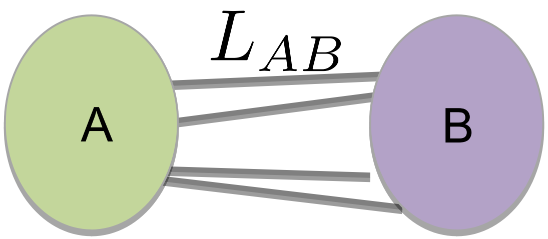
Resolution limit



$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

Resolution limit

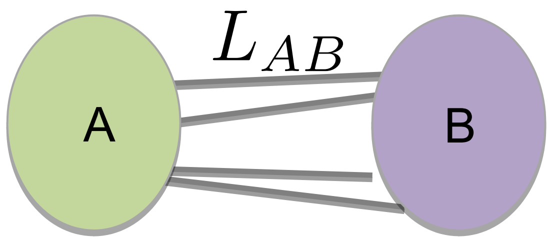


$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$

Resolution limit

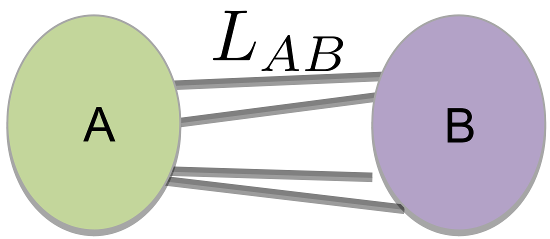


$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$ ➡ $\Delta M_{AB} > 0$ We merge A and B to maximize modularity.

Resolution limit



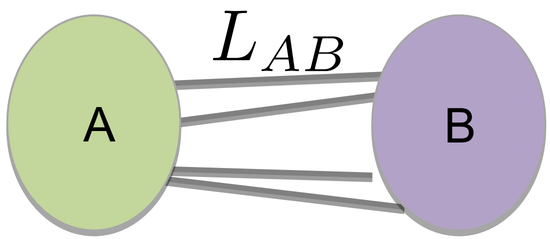
$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$ → $\Delta M_{AB} > 0$ We merge A and B to maximize modularity.

Assuming $k_A \sim k_B = k$ → $k \leq \sqrt{2L}$

Resolution limit



$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

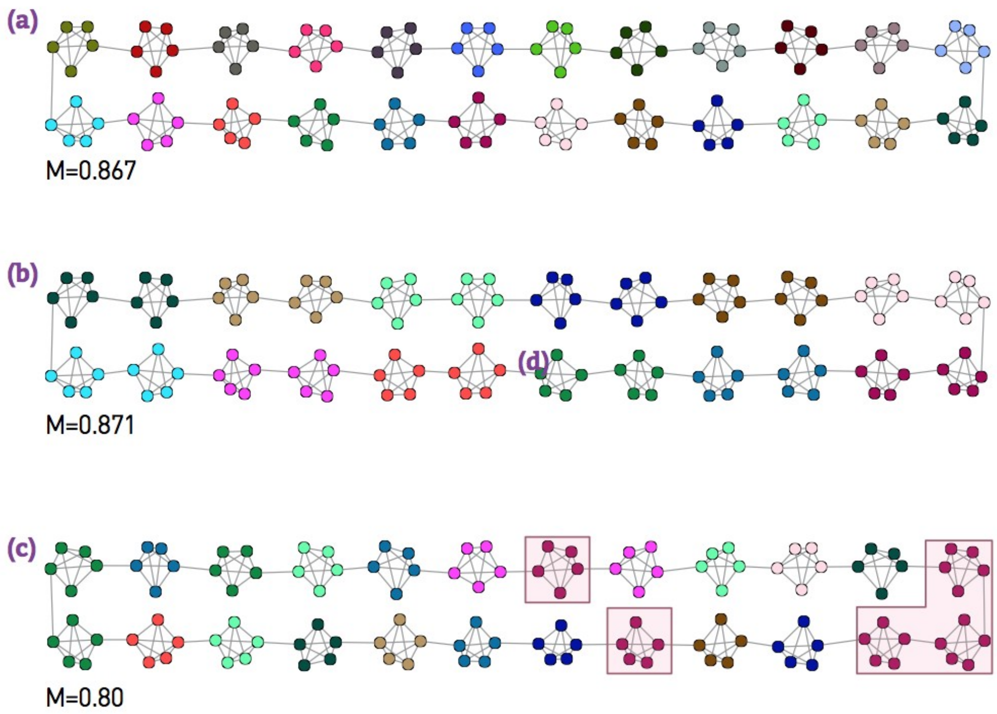
k_A and k_B total degree in A and B

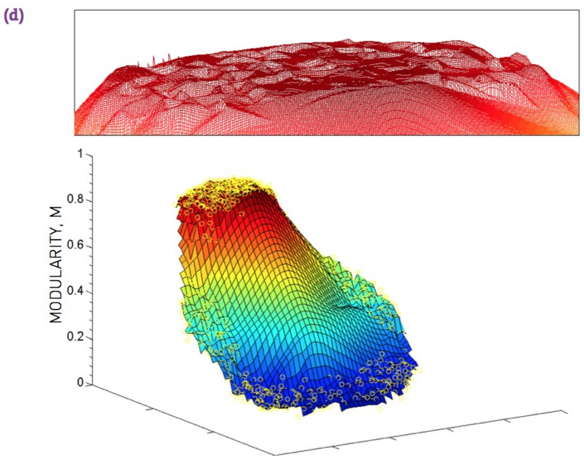
If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$ ⇒ $\Delta M_{AB} > 0$ We merge A and B to maximize modularity.

Assuming $k_A \sim k_B = k$ ⇒ $k \leq \sqrt{2L}$

Modularity has a resolution limit, as it cannot detect communities smaller than this size.

One maximum?





Null models

$$M(C_c) = \frac{1}{2L}\sum_{i,j=1}^{N}\left(A_{ij} - p_{ij}\right)\delta\left(C_i - C_j\right)$$

Expected connections
in a random model

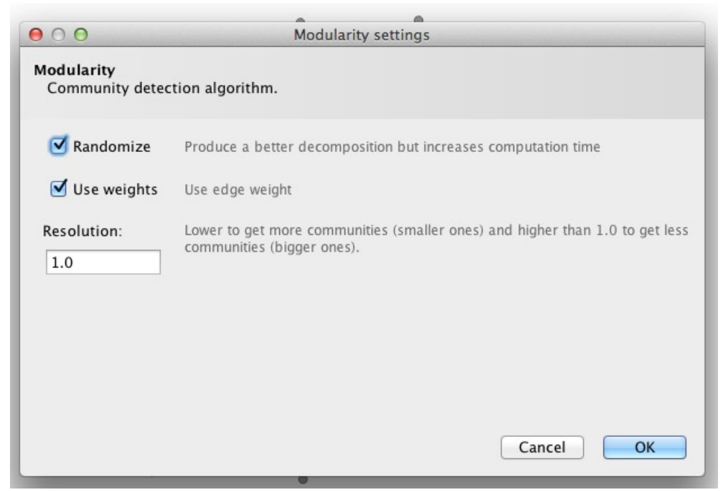
→ p_{ij} can take into account weights — S. Fortunato, *Phys. Rep.* 486 (2010)

→ p_{ij} can take into account directions — S. Fortunato, *Phys. Rep.* 486 (2010)

→ p_{ij} can take into account attributes or space — P. Expert el al., *PNAS* 108 (2011)

Section 5 Online Resources (Modularity)

Gephi



R assigns self-loops to nodes to increase or decrease the aversion of nodes to form communities

NetworkX

`community.best_partition(graph, partition=None)` ¶

Compute the partition of the graph nodes which maximises the modularity (or try..) using the Louvain heuristices

This is the partition of highest modularity, i.e. the highest partition of the dendogram generated by the Louvain algorithm.

Finds the partition that maximizes modularity (considers weights and direction)

`community.modularity(partition, graph)`

Compute the modularity of a partition of a graph

Calculates the modularity of the partition you provide

Online Resources (1)

The greedy algorithm is neither particularly fast nor particularly successful at maximizing M.

Scalability: Due to the sparsity of the adjacency matrix, the update of the matrix involves a large number of useless operations. The use of data structures for sparse matrices can decrease the complexity of the computational algorithm to , which allows us to analyze is of networks up to nodes. See **"Fast Modularity" Community Structure Inference Algorithm** http://cs.unm.edu/~aaron/research/fastmodularity.htm for the code.

A fast greedy algorithm was proposed by Blondel and collaborators, that can process networks with millions of nodes. For the description of the algorithm see **Louvain method: Finding communities in large networks** https://sites.google.com/site/findcommunities/ for the code.