Quantifying your complex network

Wang Lu
https://github.com/wanglu2014
content

• Centrality to measure Node
  • Degree、betweenness、closeness
• cliquishness圈子度 to measure Community
• Entropy to measure Network
Centrality of node

如何度量节点的重要性
• degree
• betweenness
• closeness
shortest path
degree centrality

- Degree: node with higher degree, is connected with more neighbors. Might be a "hubs"
- Degree distribution
Betweenness centrality

- Betweenness: Nodes can be connected with path. Node with higher betweenness, have more paths pass by.
  They are “medium” to connect nodes.
  Removing highest betweenness node probably isolate community.
closeness centrality

• Closeness: Node with higher closeness, have more path to other node.
They can be “broadcasters”
• Difference between closeness and betweenness: betweenness influence the flow in a system
<table>
<thead>
<tr>
<th>指标名称</th>
<th>概念</th>
<th>比较</th>
<th>实际应用</th>
</tr>
</thead>
<tbody>
<tr>
<td>点度中心度</td>
<td>在某个点上，有多少条线</td>
<td>强调某点单独的价值</td>
<td>★作为基本点的描述</td>
</tr>
<tr>
<td>接近中心度</td>
<td>该点与网络中其他点距离之和的倒数，越大说明越在中心，越能够很快到达其他点</td>
<td>强调点在网络的价值，越大，越在中心</td>
<td>★★基本描述，用户价值</td>
</tr>
<tr>
<td>中间中心度</td>
<td>代表最短距离是否都经过该点。如果都经过说明这个点很重要，其中包括线的中心度</td>
<td>强调点在其他点之间调节能力，控制能力指数，中介调节效应</td>
<td>★★推荐算法，用户的控制力</td>
</tr>
<tr>
<td>特征向量中心度</td>
<td>根据相邻点的重要性来衡量该点的价值。首先计算邻接矩阵，然后计算邻接矩阵的特征向量。</td>
<td>强调点在网络的价值，并且比接近中心度厉害的是，点价值是根据近邻点来决定的</td>
<td>★★★推荐算法，用户潜在价值</td>
</tr>
</tbody>
</table>

https://blog.csdn.net/sinat_26917383/article/details/51443846
```r
# use sna package to calculate node-level centralities, make sure input is in network format
node.centrality<-function(x){
  require(sna)
  central.nodes<-cbind(degree(x, cmode="indegree"), degree(x, cmode="outdegree"),
                       betweenness(x, rescale=T),
                       closeness(x, cmode="directed", rescale=T))
  colnames(central.nodes)<-c("indegree", "outdegree", "betweenness", "closeness")
  rownames(central.nodes)<-x%>%"vertex.names"
  list(central.nodes)
}

library(sna)
library(network)
one_mode_mat<- as.matrix(read.csv("L:/Downloads/sna_exampler/data/one_mode_ad_mat.csv", row.names=1))
# use network format to use node.centrality function
one_mode_net=network(one_mode_mat)
node.centrality(one_mode_net)
```

https://github.com/wanglu2014/sna_exampler
community

- 邻接矩阵
- 聚集系数
- 最大连通分支
邻接矩阵

<table>
<thead>
<tr>
<th></th>
<th>Andy</th>
<th>Bill</th>
<th>Carol</th>
<th>Dan</th>
<th>Ele</th>
<th>Fra</th>
<th>Gar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andy</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Bill</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Carol</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dan</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Elena</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frank</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Garth</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Their neighbors

small.network <- matrix(c(0,1,0,1,0,1,0,1,0,0,0,1,0,0,1,0,1,0,0,1,0,1,0,0,0,1,0,0,1,0,0,1,0), nrow = 6, byrow = TRUE)

• A sum of rows is simply the total number of followers a person has:
  apply(small.network, 1, sum)

• second degree followers (that is their followers are followed)
  apply((small.network %*% small.network + small.network), 1, sum)
Clustering Coefficient

- Measures the cliquishness of a particular node
  - A node is cliquish if its neighbors know each other

- 你的朋友，相互也是朋友。（朋友的朋友就是朋友）

We measure the number of triangles around a particular vertex

Clustering Coef of the black node = 0

Clustering Coef = 1

Assortativity: 度比较多的聚在一起
例子：乔丹做董事
# use igraph to calculate network-level measures, make sure input is in igraph format

```r
network.measure<-function(x){
  require(igraph)
  graph.measure<-cbind(as.numeric(length(V(x))),
                       as.numeric(length(E(x))),
                       as.numeric(diameter(x)),
                       as.numeric(graph.density(x)),
                       as.numeric(reciprocity(x)),
                       as.numeric(transitivity(x)),
                       as.numeric(average.path.length(x, unconnected=FALSE)), # important
                       as.numeric(mean(graph.strength(x))))
  colnames(graph.measure)<-c("nodes","edges","diameter",
                           "density","reciprocity","transitivity",
                           "average path length","average weighted degree")
  rownames(graph.measure)<-"x"
  list(graph.measure)
}
```

# use igraph format within network.measure function
```
data.frame(network.measure(one_mode_graph))
```

```
   nodes edges diameter  density reciprocity transitivity average.path.length
x   4    16      1  1.333333       1         1                1
x  average.weighted.degree
     8
```

```r
myPath <- 'C:/Users/Administrator/Documents/pic017/'
graphList <- list()
degreeList <- list()
attrList <- list()
for (file in list.files(pattern = '*.gml', myPath)){
  G1 <- read_graph(paste0(myPath, file), format = c("gml"))
  E(G1)$weight <- abs(E(G1)$weight)
  graphList[[file]] <- G1
degreeList[[file]] <- degree(graphList[[file]])
  attrList[[file]] <- list(mean(degree(G1)),
                          average.path.length(G1),
                          transitivity(G1), # 聚集系数
                          mean(betweenness(G1)),
                          mean(closeness(G1)))
}
attrTable <- do.call('rbind', attrList)
colnames(attrTable) <- c('mean degree', 'average path', 'transitivity', 'mean betweenness', 'mean closeness')
degreeTable <- do.call(gpcR:::cbind.na, degreeList)
write.csv(degreeTable, 'degreeTable_all.csv')
```

```r
library(ggsignif)
comparedList <- list(c("Chongming", "Shanghai"))
ggplot(data = attr_meta, mapping = aes(x=Subgroup, y=mean.degree)) +
  geom_violin(alpha = 0) +
  stat_summary(fun.y=mean, colour="darkred", geom="point", hape=18, size=3,
               show_guide = FALSE) +
  geom_jitter(alpha = 0.8, color = "tomato")+
  geom_signif(comparisons = comparedList, test=t.test, map_signif_level = TRUE,
              textsize=3, step_increase = 0.2)
```

https://github.com/wanglu2014/
Their network structure is significant different

Degree: node with higher degree, is connected with more neighbor. Might be a “hub”.

Betweenness: Nodes can be connected with path. Node with higher betweenness, have more paths pass by. They are “medium” to connect nodes.

Closeness: Node with higher closeness, have more path to other nodes. They can be “approached” by all.
信息熵

- 信息熵的度量

信息熵的总值等于每一种可能性的概率值与其对数的乘积的加和，并且取最终值的负数

\[ H(U) = E[-\log p_i] = -\sum_{i=1}^{n} p_i \log p_i \]
可是在一个非洲的原始部落里，存在着一种叫做“鼓语”的语言，这种语言不仅可以让人面对面交流，也可以通过击鼓的方式传播出去!这种语言极具韵律性，比如发音相同的两个词，但是根据声调不同，表达的意思就可能会完全不同!击鼓传播出去的声音是将发音的变化刨除，只传播声调的变化。但是这样就造成了极大的歧义性，因此，村里的鼓手想出了一个办法，那就是用尽量多的、重复的语言，描述同一件事。比如，正常说话时所说的:“所有男人们晚上到河边开会。”这样一句话，到了鼓手通过击鼓传播出去时，就变成了:“所有成年的、这个伟大村落里的男性们啊，在黄昏到来之时，在夜幕降临之际，我们相约到蜿蜒的、流淌着的河流旁边，共同谈论事情。在黄昏之时，所有的男人们都要来，来到那河边。”
网络熵

• 网络熵的计算
• 网络熵的应用举例：把性质均一的网络作为null model，然后比较真实网络与null model在该性质上，有没有熵的显著差异。

https://journals.aps.org/pre/abstract/10.1103/PhysRevE.96.012308#fulltext

• 缺点：这个熵只是一个生物系统的简单描述，丢失了许多信息。

代码
```r
# 假设g已经是一个图
# entropy 1.3
# entropy 0
```
鲁棒性(robustness)

- 依次序删去一些节点，然后度量网络整体扰动的程度
- 次序

<table>
<thead>
<tr>
<th>Degree Centrality</th>
<th>A node is important if it has many neighbors, or, in the directed case, if there are many other nodes that link to it, or if it links to many other nodes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closeness Centrality</td>
<td>mean distance from a vertex to other vertices,</td>
</tr>
<tr>
<td>Betweenness centrality</td>
<td>lies on paths between other vertices</td>
</tr>
</tbody>
</table>

图中下降越慢，受扰动越小越鲁棒
鲁棒性公式  \( R = \frac{1}{N} \sum_{i=1}^{N} \sigma(i/N) \) 

最大连接分支的比例

来源：<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0059613>
使用 QuACN 度量复杂网络

- 有特别功能的R包：第一个大规模整合了网络结构的度量指标

<table>
<thead>
<tr>
<th>QuACN</th>
<th>Descriptors based on distances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wiener Index</td>
</tr>
<tr>
<td></td>
<td>Harary Index</td>
</tr>
<tr>
<td></td>
<td>Balaban J Index</td>
</tr>
<tr>
<td></td>
<td>Mean Distance Deviation</td>
</tr>
<tr>
<td></td>
<td>Compactness</td>
</tr>
<tr>
<td></td>
<td>Product of Row Sums index</td>
</tr>
<tr>
<td></td>
<td>Hyper-distance-path index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptors other invariants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index of total adjacency</td>
</tr>
<tr>
<td>Zagreb group indices</td>
</tr>
<tr>
<td>Randić index</td>
</tr>
<tr>
<td>The complexity index B</td>
</tr>
<tr>
<td>Normalized edge complexity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classical entropy based descriptors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topological information content</td>
</tr>
<tr>
<td>Bonchev - Trinajstić indices</td>
</tr>
<tr>
<td>BERTZ complexity index</td>
</tr>
<tr>
<td>Radial centric information index</td>
</tr>
<tr>
<td>Vertex degree equality-based index</td>
</tr>
<tr>
<td>Balaban-like information indices</td>
</tr>
<tr>
<td>Graph vertex complexity index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parametric graph entropy measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>with information functionals using:</td>
</tr>
<tr>
<td>the j-spheres</td>
</tr>
<tr>
<td>path lengths</td>
</tr>
<tr>
<td>vertex centrality</td>
</tr>
</tbody>
</table>
如何构建网络

- 只需要两个（实际上也可以一个）参数

### 表3 每个函数的常见参数

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>graphNEL</td>
<td>The graph that represents the network.</td>
<td>yes</td>
</tr>
<tr>
<td>dist</td>
<td>matrix</td>
<td>The distance matrix of g, if this parameter remains empty or is set to NULL, the distance matrix will be calculated separately within the corresponding R-function.</td>
<td>no</td>
</tr>
</tbody>
</table>

This table shows the two parameters that are common for every method.

### 例子

```r
> g = new("graphNEL")
> # add nodes
> g = addNode("1", g)
> g = addNode("2", g)
> g = addEdge("1","2",g,l)
> balabanlike1(g)
[1] Inf
Warning message:
In balabanlike1(g): Graphs with |V| < 3 result in: Inf!
```

注：数据导入
library(Matrix)
library(igraph)
library(graph)
#edge matrix into igraph
G <- as.undirected(graph.adjacency(m2, weighted = T))
#igraph into graph
g <- as_graphnel(G)
Table 2 Selected descriptors for the small example graphs

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiener index $W$</td>
<td>56.000</td>
<td>52.000</td>
<td>48.000</td>
<td>44.000</td>
<td>42.000</td>
<td>36.000</td>
</tr>
<tr>
<td>Balaban-like index $X$</td>
<td>0.5979</td>
<td>0.6932</td>
<td>0.8190</td>
<td>1.0492</td>
<td>1.1452</td>
<td>1.8204</td>
</tr>
<tr>
<td>Topological information content $I_{top}$</td>
<td>1.9502</td>
<td>2.5216</td>
<td>1.3788</td>
<td>1.9502</td>
<td>1.8424</td>
<td>0.5917</td>
</tr>
<tr>
<td>Dehmer entropy $I_{\gamma}$</td>
<td>2.7648</td>
<td>2.7533</td>
<td>2.7432</td>
<td>2.7282</td>
<td>2.7305</td>
<td>2.7391</td>
</tr>
</tbody>
</table>

Results of some selected descriptors applied to the small example graphs shown in Figure 2.
去掉特定类型的节点：
- 例如：去掉它，再去掉它的邻居
- [https://github.com/dnarango/Keystone-Plants](https://github.com/dnarango/Keystone-Plants)
- 例如：去掉一类节点
- [https://github.com/wanglu2014/robustness](https://github.com/wanglu2014/robustness)
SnapVX - A convex optimization solver for problems defined on a graph.

TQ (Temporal Quantities) - Python 3 library for temporal network analysis.

R

For more awesome R resources, see the Awesome R and Awesome R Books lists. See also this Google spreadsheet by Ian McCulloh and others.

To convert many different network model results into tidy data frames, see the broom package. To convert many different network model results into LaTeX or HTML tables, see the texreg package.

amen - Additive and multiplicative effects models for relational data.

Bergm - Tools to analyse Bayesian exponential random graph models (BERGM).

bipartite - Functions to visualize bipartite networks and compute indices commonly used in ecological research.

blockmodeling - Implementats generalized blockmodeling for valued networks.

bnlearn - Tools for Bayesian network learning and inference (related Shiny app).

btergm - Tools to fit temporal ERGMs by bootstrapped pseudolikelihood. Also provides MCMC maximum likelihood estimation, goodness of fit for ERGMs, TERGMs, and stochastic actor-oriented models (SAOMs), and tools for the micro-level interpretation of ERGMs and TERGMs.

https://github.com/briatte/awesome-network-analysis
谢谢
Giant component and robustness

- Giant component 最大连通分支
- Robustness 的度量指标
- $u = 1 - \frac{N_G}{N}$
Topological Information Content

$$P \text{值} = 0.015$$

Topological Information Content \([17, 18]\):

$$I_{\text{orb}}^V(G) := -\sum_{i=1}^{k} \frac{|N_i^V|}{N} \log \left( \frac{|N_i^V|}{N} \right).$$

$$|N_i^V|$$ denotes the number of vertices belonging to the \(i\)-th vertex orbit.
Vertex Orbits and Edge Orbits

Def 2.4. The equivalence classes of the vertices of a graph $G$ under the action of the automorphisms are called vertex orbits. The equivalence classes of the edges are called edge orbits.

Example 2.10.

vertex orbits: $\{1,8\}$, $\{4,6\}$, $\{2,7\}$, $\{3,5\}$
edge orbits: $\{12,78\}$, $\{34,56\}$, $\{23,25,37,57\}$, $\{35\}$

$I_{2a} = \log_2 3$
$= 1.58$ bit.

$I_{2b} = 0$.

Automorphisms

$\lambda_0 = (1)(2)(3)(4)(5)(6)(7)(8)$
$\lambda_1 = (1\ 8)(2\ 7)(3)(4)(5)(6)$
$\lambda_2 = (1)(2)(3\ 5)(4\ 6)(7)(8)$
$\lambda_3 = (1\ 8)(2\ 7)(3\ 5)(4\ 6)$
<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>R function</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Desaters based on distances</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skorobogatov indices</td>
<td>D(G)</td>
<td>dobrynin(g)</td>
<td>[37]</td>
</tr>
<tr>
<td>Wiener index</td>
<td>W(G)</td>
<td>wiener(g)</td>
<td>[36]</td>
</tr>
<tr>
<td>Harary index</td>
<td>H(G)</td>
<td>harary(g)</td>
<td>[53]</td>
</tr>
<tr>
<td>Randić J index</td>
<td>J(G)</td>
<td>balabanJ1(g)</td>
<td>[54]</td>
</tr>
<tr>
<td>Compactness</td>
<td>C(G)</td>
<td>compactness(g)</td>
<td>[55]</td>
</tr>
<tr>
<td>Product of row sums index</td>
<td>PRS(G)</td>
<td>productOfRowSums(g)</td>
<td>[56]</td>
</tr>
<tr>
<td>Hyper-distance-path index</td>
<td>D2(G)</td>
<td>hyperDistancePathIndex(g)</td>
<td>[24]</td>
</tr>
<tr>
<td><strong>Desaters based on other invariants</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index of total adjacency</td>
<td>A(G)</td>
<td>totalAdjacency(g)</td>
<td>[39]</td>
</tr>
<tr>
<td>Zagreb group indices 1</td>
<td>Z1(G)</td>
<td>zageb1(g)</td>
<td>[38]</td>
</tr>
<tr>
<td>Zagreb group indices 2</td>
<td>Z2(G)</td>
<td>zageb2(g)</td>
<td>[38]</td>
</tr>
<tr>
<td>Randić index</td>
<td>R(G)</td>
<td>randic(g)</td>
<td>[57]</td>
</tr>
<tr>
<td>The complexity index B</td>
<td>B(G)</td>
<td>complexityIndexB(g)</td>
<td>[39]</td>
</tr>
<tr>
<td>Normalized edge complexity</td>
<td>E(G)</td>
<td>normalizedEdgeComplexity(g)</td>
<td>[39]</td>
</tr>
<tr>
<td><strong>Classical entropy-based descriptors</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topological information content</td>
<td>I_e(G)</td>
<td>topologicalInfoContent(g)</td>
<td>[42]</td>
</tr>
<tr>
<td>Bonchev-Trinajstić index 1</td>
<td>I_a(G)</td>
<td>bonchev1(g)</td>
<td>[42]</td>
</tr>
<tr>
<td>Bonchev-Trinajstić index 2</td>
<td>I_b(G)</td>
<td>bonchev2(g)</td>
<td>[42]</td>
</tr>
<tr>
<td>BERT2 complexity index</td>
<td>C(G)</td>
<td>bertz(g)</td>
<td>[58]</td>
</tr>
<tr>
<td>Radial centric info index</td>
<td>c(G)</td>
<td>radialCentric(g)</td>
<td>[20]</td>
</tr>
<tr>
<td>Vertex degree equality-based II</td>
<td>I_m(G)</td>
<td>vertexDegree(g)</td>
<td>[20]</td>
</tr>
<tr>
<td>Balaban-like information index U</td>
<td>U(G)</td>
<td>balabanlike1(g)</td>
<td>[40]</td>
</tr>
<tr>
<td>Balaban-like information index X</td>
<td>X(G)</td>
<td>balabanlike2(g)</td>
<td>[40]</td>
</tr>
<tr>
<td>Graph vertex complexity index</td>
<td>I_v(G)</td>
<td>graphVertexComplexity(g)</td>
<td>[59]</td>
</tr>
</tbody>
</table>

This table gives an overview about the implemented topological network descriptors including the function name in QuACN and the reference to the corresponding publication.
综合性的例子

```r
> sapply(glist, balabanlike2)
```

(a) ![Diagram](image1.png)

(b) ![Diagram](image2.png)

(c) ![Diagram](image3.png)

(d) ![Diagram](image4.png)

(e) ![Diagram](image5.png)

(f) ![Diagram](image6.png)
综合性的例子

<table>
<thead>
<tr>
<th>Sample Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
</tr>
<tr>
<td>(b)</td>
</tr>
<tr>
<td>(c)</td>
</tr>
<tr>
<td>(d)</td>
</tr>
<tr>
<td>(e)</td>
</tr>
<tr>
<td>(f)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Normalized Descriptor Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.8</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 2 Selected descriptors for the small example graphs

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weiner index $W$</td>
<td>56.000</td>
<td>52.000</td>
<td>48.000</td>
<td>44.000</td>
<td>40.000</td>
<td>36.000</td>
</tr>
<tr>
<td>Balaban-like index $X$</td>
<td>0.5979</td>
<td>0.6992</td>
<td>0.8190</td>
<td>1.0492</td>
<td>1.1452</td>
<td>1.8204</td>
</tr>
<tr>
<td>Topological Information content $I_{	ext{top}}$</td>
<td>1.9502</td>
<td>2.5216</td>
<td>1.3788</td>
<td>1.9502</td>
<td>1.8424</td>
<td>0.5917</td>
</tr>
<tr>
<td>Dehmer entropy $I_{	ext{Deh}}$</td>
<td>2.7648</td>
<td>2.7533</td>
<td>2.7432</td>
<td>2.7282</td>
<td>2.7305</td>
<td>2.7391</td>
</tr>
</tbody>
</table>

Results of some selected descriptors applied to the small example graphs shown in Figure 2.
we extracted subgraphs from benign and cancer networks based on the gene ontology (GO) database.

For each network and each GO-term we extracted one subgraph containing the genes associated with this specific GO-term.

This resulted in a total of 159 networks representing benign tissue and 108 networks representing cancer tissue.

Calculated all topological network descriptors available in QuACN, as feature vectors for each of these networks. Afterwards, we performed feature selection and classification using random forest with 10-fold cross-validation (CV).

\[
F\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

F-score of 0.80

Accuracy of 0.74
谢谢
Natural Connectivity

- **spiec-easi** is a calculation-related R package that can construct a related network based on sparse matrices.
- Its author refers to **Natural Connectivity** to compare network connectivity.
- A special case of the average eigenvalue

\[
\bar{\lambda} = \ln \left( \frac{1}{N} \sum_{i=1}^{N} e^{\lambda_i} \right),
\]

(3)

\[
0 \leq \bar{\lambda} \leq N - \ln N.
\]

(5)

```r
spiec.out <- spiec.easi(phyloseqobj, method='mb',
lambda.min.ratio=1e-2, nlambda=20,
icov.select.params=list(rep.num=20))
spiec.graph <- adj2igraph(spiec.out$refit,
vertex.attr=list(name=taxa_names(phyloseqobj)))
natcon <- function(ig) {
    N <- vcount(ig)  # Order (number of vertices) of a graph
    adj <- get.adjacency(ig)
    evals <- eigen(adj)$value
    nc <- log(mean(exp(evals)))
    nc / (N - log(N))
    nc.attack <- function(ig) {
        hubord <- order(rank(betweenness(ig)),
                        rank(degree(ig)), decreasing=TRUE)
        sapply(1:round(vcount(ig)*.8), function(i) {
            ind <- hubord[1:i]
            tmp <- delete_vertices(ig, V(ig)$name[ind])
            natcon(tmp)})
    }
    nc_name<-paste0("nc",name,threshod)
    lsnc[[nc_name]] <- nc.attack(spiec.graph)
}
lsnc_frame<-do.call("rbind", lsnc)
```
Natural Connectivity

- **spiec-easi** is a package in R for computing sparsity of networks, which can be used to construct networks based on sparse matrices.
- Its authors refer to **Natural Connectivity** to compare the connectivity of networks.
- **Understanding Network Concepts in Modules**

A network is defined by the set of $V$ vertices (nodes, points), $\{V\} = \{v_1, v_2, \ldots, v_V\}$, and the set of $E$ edges (links, lines), $\{E\} = \{E_1, E_2, \ldots, E_E\}$. The edge $\{ij\}$ is the line that emanates from vertex $i$ and ends in vertex $j$. A subgraph is a graph obtained from the parent graph by deleting at least one edge or a vertex with its incident edges. A loop is an edge that begins and ends in the same vertex. A multigraph is a graph in which some pairs of vertices are linked by more than one edge. Simple graphs are graphs having no multiple edges and loops. In a complete graph, $K_V$, any two vertices are connected by an edge. A directed graph is a graph having at least one directed edge. Directed edges are termed arcs. Graph without any directed edge is undirected. The graph is connected when there is a path between any pair of vertices in it; otherwise the graph is
disconnected. A path in the graph is a sequence of adjacent edges without traversing any vertex twice. A path graph, $P_v$, is a graph containing only one path. A star-graph, $S_v$, is a graph containing one central vertex and $V-1$ branches of length one edge. A walk is a alternating sequence of vertices and edges, each of which could be traversed more than once. The walk length is the number of edges in it. A cycle is a path that starts from and ends in the same vertex. Graphs containing at least one cycle are called cyclic graphs.

Trees are graphs containing no cycles. A spanning tree is a connected acyclic graph containing all the vertices of the graph. Graph components are connected subgraphs of vertices that are not connected to each other. Euler’s theorem relates the number of
Content

• Here we study network concepts in special types of networks, which we refer to as approximately factorizable networks. In these networks, the pairwise connection strength (adjacency) between 2 network nodes can be factored into node specific contributions, named node 'conformity'.

• Scope: Our results apply to modules in gene co-expression networks and to special types of modules in protein-protein interaction networks
Background

• Network concepts are also known as network statistics or network indices
  – Examples: connectivity (degree), clustering coefficient, topological overlap, etc

• Network concepts underlie network language and systems biological modeling.

• Dozens of potentially useful network concepts are known from graph theory.

• **Question**: How are seemingly disparate network concepts related to each other?
Review of *some* fundamental network concepts
Connectivity

• Gene connectivity = row sum of the adjacency matrix
  - For unweighted networks = number of direct neighbors
  - For weighted networks = sum of connection strengths to other nodes

\[ \text{Connectivity}_i = k_i = \sum_{j \neq i} a_{ij} \]
Density

• Density = mean adjacency
• Highly related to mean connectivity

\[
Density = \frac{\sum_{i} \sum_{j \neq i} a_{ij}}{n(n-1)} = \frac{S_1(k)}{n(n-1)} = \frac{mean(k)}{n-1}
\]

where \( n \) is the number of network nodes.
Centralization

\[
Centralization = \frac{n}{n-2} \left( \frac{\max(k)}{n-1} - \text{Density} \right) \approx \frac{\max(k)}{n-1} - \text{Density}
\]

= 1 if the network has a star topology

= 0 if all nodes have the same connectivity

Centralization = 1 because it has a star topology

Centralization = 0 because all nodes have the same connectivity of 2
Heterogeneity

- Heterogeneity: coefficient of variation of the connectivity
- Highly heterogeneous networks exhibit hubs

\[
\text{Heterogeneity} = \frac{\sqrt{\text{variance}(k)}}{\text{mean}(k)}
\]
Clustering Coefficient

Measures the cliquishness of a particular node « A node is cliquish if its neighbors know each other »

\[ \text{ClusterCoef}_i = \frac{\sum_{l \neq i} \sum_{m \neq i, l} a_{il}a_{lm}a_{mi}}{\left( \sum_{l \neq i} a_{il} \right)^2 - \sum_{l \neq i} a_{il}^2} \]

This generalizes directly to weighted networks (Zhang and Horvath 2005)

Clustering Coef of the black node = 0

Clustering Coef = 1
The topological overlap dissimilarity is used as input of hierarchical clustering.

\[ TOM_{ij} = \sum_{u \neq i, j} a_{iu} a_{uj} + a_{ij} \]

\[ DistTOM_{ij} = 1 - TOM_{ij} \]

- Generalized in Zhang and Horvath (2005) to the case of weighted networks
- Generalized in Yip and Horvath (2007) to higher order interactions
- Generalized in Li and Horvath (2006) to multiple nodes
Question: What do all of these fundamental network concepts have in common?

Answer: They are tensor valued functions of the off-diagonal elements of the adjacency matrix A.
CHALLENGE

Challenge: Find relationships between these and other seemingly disparate network concepts.

• For general networks, this is a difficult problem.
• But a solution exists for a special subclass of networks: approximately factorizable networks
• Motivation:
  modules in larger networks are often approximately factorizable
Approximately factorizable networks and conformity

We define an adjacency matrix $A$ to be exactly factorizable if, and only if, there exists a vector $CF$ with non-negative elements such that

$$a_{ij} = CF_i CF_j \quad \text{for all} \quad i \neq j$$

We also define the concept of conformity for a general, non-factorizable network. Idea: approximate $A$ with an exactly factorizable adjacency matrix

$$A_{CF} = CF CF^T - \text{diag}(CF^2) + I$$

We define the conformity as a maximizer of the factorizability function

$$F_A(v) = 1 - \frac{\sum_i \sum_{j \neq i} (a_{ij} - v_i v_j)^2}{\sum_i \sum_{j \neq i} (a_{ij})^2}$$
The conformity vector reduces the dimensionality of the adjacency matrix

- Note that the (symmetric) adjacency matrix contains $n^*(n-1)/2$ parameters $a(i,j)$.
- The conformity vector contains only $n$ parameters $CF(i)$.
- Thus, by focusing on the conformity based adjacency matrix, we effectively reduce the dimensionality of the adjacency matrix.
- This approximation is only valid if the network has high factorizability as defined on the next slide.
The higher $F(A)$, the better $A_{CF}$ approximates $A$

- The factorizability $F(A)$ is normalized to take on values in the unit interval $[0, 1]$.

$$F(A) = 1 - \frac{\| (A - I) - (A_{CF} - I) \|_F^2}{\| A - I \|_F^2}$$

Empirical observation: subnetworks comprised of module genes tend to have high factorizability $F(A) > 0.8$
Applications: modules in
a) protein-protein networks
b) gene co-expression networks
The Topological Overlap Matrix Can Be Considered as Adjacency Matrix

• Important insight for protein-protein interaction (PPI) networks:
  • Since the matrix $\text{TopOverlap}[i,j]$ is symmetric and its entries lie in $[0, 1]$, it satisfies our assumptions on an adjacency matrix.
  • Since the adjacency matrices of our PPI networks are very sparse, we replaced them by the corresponding topological overlap matrices.
  • Roughly speaking, the topological overlap matrix can be considered as a 'smoothed out' version of the adjacency matrix.
Hierarchical clustering dendrogram and module definition.
Drosophila PPI network.

The color-band below the dendrogram denotes the modules, which are defined as branches in the dendrogram. Of the 1371 proteins, 862 were clustered into 28 proper modules, and the remaining proteins are colored in grey;

Recall that we used TOM instead of the original adjacency matrix as weighted network between the proteins.
Hierarchical clustering dendrogram and module definition.

Yeast PPI network
Yeast Co-expression Network: Soft Thresholding

Colored by module membership
Observation 1

- Sub-networks comprised of module nodes tend to be approximately factorizable.
- Specifically, they have high factorizability $F(A)$

Table 1: Summary of fundamental network concepts in real network applications.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Fly Protein</th>
<th>Yeast Protein</th>
<th>Yeast (Weighted)</th>
<th>Yeast (Unweighted)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proper</td>
<td>Grey</td>
<td>Proper</td>
<td>Grey</td>
</tr>
<tr>
<td>Factorizability</td>
<td>.82 (.086)</td>
<td>.170</td>
<td>.85 (.100)</td>
<td>.200</td>
</tr>
<tr>
<td>Density</td>
<td>.21 (.074)</td>
<td>.017</td>
<td>.28 (.120)</td>
<td>.026</td>
</tr>
<tr>
<td>Centralization</td>
<td>.18 (.091)</td>
<td>.052</td>
<td>.20 (.055)</td>
<td>.036</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>.35 (.130)</td>
<td>.460</td>
<td>.36 (.140)</td>
<td>.430</td>
</tr>
<tr>
<td>Mean Cluster Coef.</td>
<td>.28 (.110)</td>
<td>.050</td>
<td>.36 (.120)</td>
<td>.093</td>
</tr>
<tr>
<td>Mean Conformity</td>
<td>.45 (.076)</td>
<td>.130</td>
<td>.51 (.120)</td>
<td>.150</td>
</tr>
</tbody>
</table>
We use both PPI and gene co-expression network data to show empirically that subnetworks comprised of module nodes are often approximately factorizable.

CAVEATS

• Approximate factorizability is a very stringent structural assumption that is not satisfied in general networks.
• Modules in gene co-expression networks tend to be approximately factorizable if the corresponding expression profiles are highly correlated,
• the situation is more complicated for modules in PPI networks: only after replacing the original adjacency matrix by a 'smoothed out' version (the topological overlap matrix), do we find that the resulting modules are approximately factorizable.
To reveal relationships between network concepts, we use a trick.

We focus attention to the approximate conformity based adjacency matrix.

\[ A_{CF,app} = CFCF^\tau = [CF_i CF_j] \]

- Strictly speaking it violates our assumption on an adjacency matrix since its diagonal elements are not 1.
- It is very useful for defining approximate conformity based network concepts.
- Approximately conformity based network concepts have several theoretical advantages as we detail below.
Network Concept Functions

Abstract definition:
tensor-valued function of a general $n \times n$ matrix $M = [m_{ij}]$ a general matrix.

Examples

\[ Connectivity_i(M) = \sum_j m_{ij} = e_i^r M 1, \]

\[ Density(M) = \frac{\sum_i \sum_j m_{ij}}{n(n-1)}, \]

\[ Centralization(M) = \frac{n}{n-2} \left( \frac{\max(M 1)}{n-1} - Density(M) \right), \]

\[ Heterogeneity(M) = \sqrt{\frac{n(1^r M M 1)}{(1^r M 1)^2} - 1}, \]

\[ TopOverlap_{ij}(M) = \frac{e_i^r M M e_j + e_j^r M e_j}{\min\{e_i^r M 1, e_j^r M 1\} + 1 - e_i^r M e_j}, \]

\[ ClusterCoef_i(M) = \frac{e_i^r M M M e_i}{e_i^r M B_M M e_i}, \]
Table 2: Brief overview of different types of network concepts.

<table>
<thead>
<tr>
<th>Input Matrix</th>
<th>Type of Concept</th>
<th>Example: Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A - I$</td>
<td>fundamental</td>
<td>$Connectivity_i(A - I)$ = $\sum_{j\neq i} a_{ij}$</td>
</tr>
<tr>
<td>$A_{CF} - I = \mathbf{CF} \mathbf{CF}^T - \text{diag} (\mathbf{CF}^2)$</td>
<td>CF-based</td>
<td>$Connectivity_i(A_{CF} - I)$ = $CF_i\sum_{j\neq i} CF_j$</td>
</tr>
<tr>
<td>$A_{CF,app} = \mathbf{CF} \mathbf{CF}^T$</td>
<td>approximate CF-based</td>
<td>$Connectivity_i(A_{CF,app})$ = $CF_i\sum_{j} CF_j$</td>
</tr>
</tbody>
</table>

A network concept arises by evaluating a network concept function on a special type of input matrix. We assume that the diagonal elements of the matrix $A - I$ are 0.
Question: Find simple relationships between approximate CF based network concepts

\[
\begin{align*}
    k_{CF,app,i} &= CF_i S_1(CF), \\
    \text{Density}_{CF,app} &= \frac{S_1(CF)^2}{n(n-1)} \approx \left( \frac{S_1(CF)}{n} \right)^2, \\
    \text{Centralization}_{CF,app} &= \frac{n S_1(CF)}{(n-1)(n-2)} \left( \max(CF) - \frac{S_1(CF)}{n} \right) \\
    &\approx \frac{S_1(CF)}{n} \left( \max(CF) - \frac{S_1(CF)}{n} \right), \\
    \text{Heterogeneity}_{CF,app} &= \sqrt{\frac{n S_2(CF)}{S_1(CF)^2} - 1}, \\
    \text{ClusterCoef}_{CF,app,i} &= \left( \frac{S_2(CF)}{S_1(CF)} \right)^2, \\
    \text{TopOverlap}_{CF,app,ij} &\approx \frac{CF_i CF_j \left( S_2(CF) + 1 \right)}{\min(CF_i, CF_j) S_1(CF) + 1 - CF_i CF_j}.
\end{align*}
\]
Observation 1

Major advantage of approximate CF-based network concepts: they exhibit simple relationships

Relationship between heterogeneity, density, and clustering coefficient

\[
Heterogeneity_{CF, app} \approx \sqrt{\frac{ClusterCoef_{CF, app}}{Density_{CF, app}}} - 1
\]

\[
ClusterCoef_{CF, app, i} \approx \left(1 + Heterogeneity_{CF, app}^2\right)^2 \times Density_{CF, app}
\]
Observation 2

• *Fundamental network concepts are approximately equal to their approximate CF-based analogs in approximately factorizable networks*

• Recall that fundamental network concepts are defined with respect to the adjacency matrix

• Approximate CF-based network concepts are defined with respect to the conformity vector.
Drosophila PPI module networks: the relationship between fundamental network concepts $\text{NetworkConcept}$ (y-axis) and their approximate CF-based analogs $\text{NetworkConceptCF}_{\text{app}}$ (x-axis).

$R^2=0.99$

$R^2=0.9$

$R^2=0.87$

$R^2=0.87$
Yeast PPI module networks: the relationship between fundamental network concepts *NetworkConcept* (y-axis) and their approximate CF-based analogs *NetworkConceptCF,app* (x-axis).

![Graph A](image1)

\[ R^2 = 0.99 \]

![Graph B](image2)

\[ R^2 = 0.71 \]

![Graph C](image3)

\[ R^2 = 0.88 \]

![Graph D](image4)

\[ R^2 = 0.7 \]
Yeast gene co-expression module networks: the relationship between fundamental network concepts $\text{NetworkConcept}(A-I)$ (y-axis) and their approximate CF-based analogs $\text{NetworkConceptCF,app}$ (x-axis).
Observation 3

Approximate relationships between network concepts in modules

\[ \text{mean}(\text{ClusterCoef}) \approx \left(1 + \text{Heterogeneity}^2\right)^2 \times \text{Density} \]

\[ \text{TopOverlap}_{ij} \approx \frac{\max(k_i, k_j)}{n} \times \left(1 + \text{Heterogeneity}^2\right) \]

\[ \text{TopOverlap}_{[1]j} \approx \frac{k_{[1]}}{n} \times \left(1 + \text{Heterogeneity}^2\right) \]

\[ \approx \left(\text{Centralization} + \text{Density}\right) \times \left(1 + \text{Heterogeneity}^2\right) \]

The topological overlap between two nodes is determined by the maximum of their respective connectivities and the heterogeneity.
Observation 3 (cont’d)

• The mean clustering coefficient is determined by the density and the network heterogeneity in approximately factorizable networks.

• Other examples involve the topological overlap

• Thus, seemingly disparate network concepts satisfy simple and intuitive relationships in these special but biologically important types of networks.
Drosophila PPI module networks: the relationship between fundamental network concepts.

A

\[ R^2 = 0.87 \]

B

\[ R^2 = 0.91 \]

C

D

E

F
Yeast PPI module networks: the relationship between fundamental network concepts.

\[ R^2 = 0.74 \]

\[ R^2 = 0.76 \]
Yeast gene co-expression module networks: the relationship between fundamental network concepts.

A

$\text{p}=7, \ R^2=0.82$

B

$\text{tau}=0.65, \ R^2=0.6$

C

$\text{p}=7, \ R^2=0.81$

D

$\text{tau}=0.65, \ R^2=0.74$
Observation 4: network concepts are simple function of the connectivity in approximately factorizable networks

\[
\text{ClusterCoef}_i \approx \frac{(S_2(k))^2}{(S_1(k))^3},
\]

\[
\text{TopOverlap}_{ij} \approx \frac{\max(k_i, k_j)}{n} \times \frac{S_2(k)}{S_1(k)},
\]

where the last approximation assumes

\[
\frac{S_1(k)}{S_2(k)} \approx 0 \quad \text{and} \quad \frac{S_1(k) - k_ik_j}{\min(k_i, k_j)S_1(k)} \approx 0
\]
Robustness to module definition

• In our applications, we define modules as branches of an average linkage hierarchical clustering tree based which uses the topological overlap measure as input.
• But our theoretical results are applicable to any approximately factorizable network.
• We find that the theoretical results are quite robust with respect to the underlying assumptions and are highly robust with respect to the module definition.
Summary

- We study network concepts in special types of networks, which we refer to as approximately factorizable networks.
- To provide a formalism for relating network concepts to each other, we define three types of network concepts: fundamental-, conformity-based-, and approximate conformity-based concepts.
- The approximate conformity-based analogs of fundamental network concepts have several theoretical advantages.
  1. They allow one to derive simple relationships between seemingly disparate networks concepts.
     For example, we derive simple relationships between the clustering coefficient, the heterogeneity, the density, the centralization, and the topological overlap.
  2. Approximate conformity-based network concepts is that they allow one to show that fundamental network concepts can be approximated by simple functions of the connectivity in module networks.
What is the conformity?

We find that for most real networks, the conformity is highly related to the first eigenvector of the adjacency matrix, i.e.

\[ CF(i) \propto \sqrt{d_1} u_1(i) \]

where

- \( d_1 \) is the largest singular value of \( A \)
- \( u_1 \) is the corresponding unit length eigenvector with positive components.

This insight leads to an iterative algorithm for computing \( CF \), see the next slide.
Monotonic algorithm for computing the conformity

\[ \hat{A}(i-1) = A - I + \text{diag} \left( CF(i-1)^2 \right) \]

\[ CF(i) = \sqrt{d_1(i-1) \times u_1(i-1)} \]

\[ F_A \left( CF(i) \right) \geq F_A \left( CF(i-1) \right) \]