Моделирование корреляционных связей в сообществе с помощью сетей

Автор: Владимир Шитиков
https://stok1946.blogspot.com/

1. Введение

Одним из эффективных методов анализа и визуализации корреляционных структур, характерных для экологических или социальных сообществ, является использование сетевых моделей.

Сеть (англ. *network*) представляет собой конечный ориентированный граф $G = (V, E)$, где V – множество узлов (англ. *nodes*), а $E \subset V \times V$ – множество рёбер (*edge*), определяющих связи между узлами и имеющих ненулевой вес (или "пропускную способность" – *capacity*). Обычно для многомерных количественных данных можно построить два типа сетевых моделей: корреляционную сеть и частную корреляционную сеть. Теоретически каждое ребро корреляционной сети является двунаправленным (обычно двойная направленность ребер опущена, чтобы сделать график более четким). Частная корреляционная сеть обычно обозначается неориентированными ребрами:

Как корреляционная сеть, так и частная корреляционная сеть являются насыщенными моделями, т.е. узлы полностью связаны между собой ребрами. Под выбором оптимальной модели понимается определение, какие ребра можно удалить (приравнять нулю), уменьшив тем самым число степеней свободы без значимого ущерба для информационной ценности графа. Это можно сделать в широком смысле тремя способами:

- *Установление пороговых значений*: просто удаляются все ребра, пропускная способность которых меньше заданного порога, например, все ребра, статистически значимо не отличимые от нуля.
- *Селекция оптимальной модели*: пошаговые сравнения с использованием информационных критериев для идентификации подмножества «избыточных» ребер.
- *Оценка лассо*: совместная оценка значений параметров и информативности ребер с помощью методов регуляризации.

В этом сообщении мы опишем использование основных функций пакета `qgraph` версии 1.6.4, который является не только превосходным инструментом визуализации сетей, но и содержит полный набор графоаналитических методов, включая «разрежение» взвешенных матриц данных и оценку структуры графа [Epskamp et al., 2012, 2014, 2019]

2. Корреляционные сети

Источником для построения корреляционной сети является корреляционная матрица. Отметим, что коэффициент корреляции, призванный оценить отношения между двумя произвольными узлами A и B, к сожалению, включает не только «чистую» зависимость между ними, но и условные индуцированные взаимодействия этих узлов с остальными узлами сети. В частности, если мы ожидаем, что пары узлов {A и B} и {B и C} коррелируют между собой, то одновременно будут также коррелировать A и C, причем все три коэффициента корреляции включают некоторую ассоциированную составляющую, смещающую оценку выраженности "чистых" парных связей. Однако, корреляционные сети, не столь безупречные с теоретической точки зрения, могут быть весьма интересны на практике для разведочной визуализации корреляционных паттернов в наборах данных.

Рассмотрим использование функций пакета `qgraph` на примере анализа данных гидробиологической съемки [Зинченко и др., 2018] в бассейне высокоминерализированного оз. Эльтон (Волгоградская обл.). Файл с исходными данными для проведения расчетов можно скачать по адресу: http://www.ievbras.ru/ecostat/Kiril/R/Blog/Elton_MDM.RData. Отметим, что эти данные мы подробно рассматривали при выполнении многомерного ординационного анализа в сообщении https://stok1946.blogspot.com/2019/03/blog-post.html.

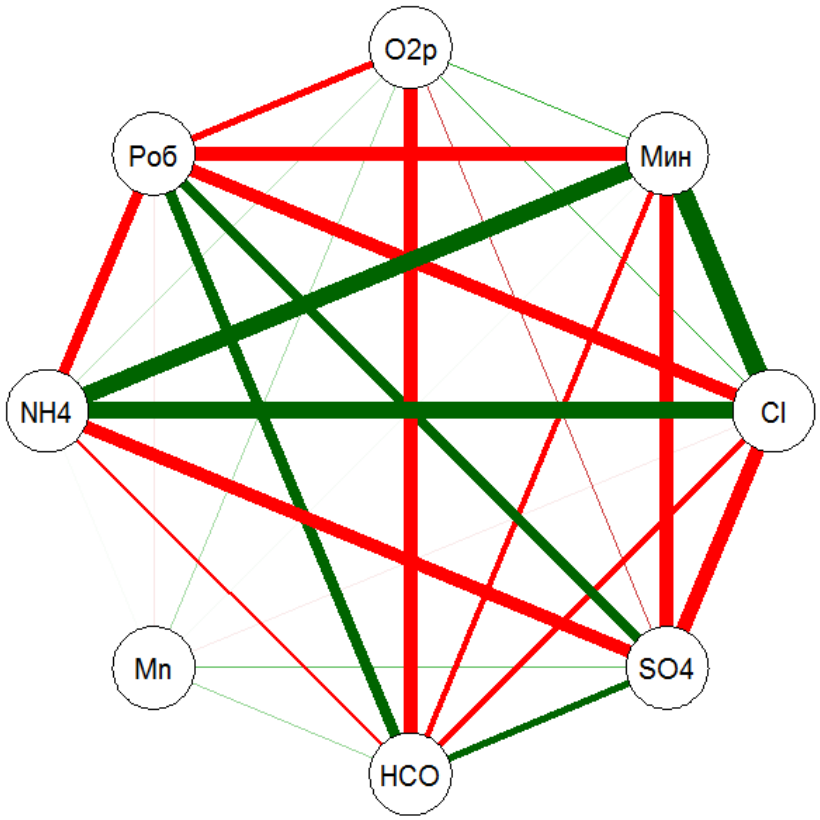
На 15 станциях наблюдений, расположенных по руслу 5 малых рек от истоков до впадения в озеро, были зарегистрированы значения 8 гидрохимических показателей, представленных в файле `Elton_MDM.RData` таблицей `Tchim.class`:

```
load(file="Elton_MDM.RData")
ls()
[1] "tab.MB"       "tab.ZB"       "tab.ZP"       "Tchim.class" "TDB"
colnames(Tchim.class)
[1] "O2p"        "Мин"        "Cl"         "SO4"        "HCO3"       "Mn"
[7] "NH4"        "Роб"        "Min.class"
```

Кроме того в состав комплекта исходных данных включены три однотипные таблицы `tab.MB`, `tab.ZB`, `tab.ZP`, описывающие отдельно видовой состав мейобентоса, зообентоса и зоопланктона соответственно. Каждая из них содержит по 15 строк и включают оценки популяционной плотности экземпляров 88 выделенных таксонов гидробионтов этих групп в баллах от 1 до 6 (0 – отсутствие вида в пробе).

Для построения корреляционной сети рассчитаем на основе данных таблицы `Tchim.class` матрицы парных корреляций Пирсона и Спирмена (поскольку существуют обоснованные сомнения относительно нормальности распределения некоторых гидрохимических показателей). Далее можно построить насыщенную корреляционную сеть, для чего просто достаточно использовать корреляционную матрицу в качестве основного аргумента функции `qgraph()`. Для управления форматом выводимого изображения и его частных атрибутов в пакете предусмотрены многочисленные опции графических параметров, добавление которых позволяет сделать диаграмму сети хорошо читаемой.

```
library(qgraph)
Cor_P <- cor(Tchim.class[,-9])
Cor_S <- cor(Tchim.class[,-9], method="spearman")
corGraph <- qgraph(Cor_P, layout = "circular", graph = "cor",
       cut = 0.3, maximum = 1, minimum = 0)
```

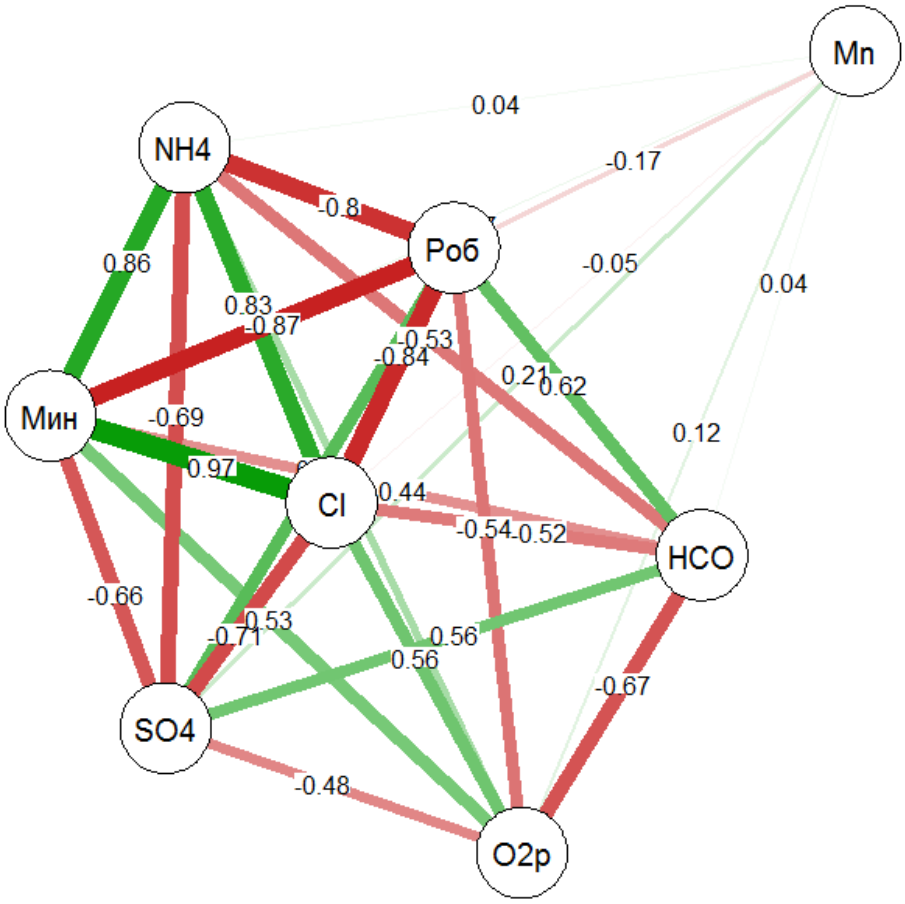


Аргумент `graph = "cor"` сообщает функции `qgraph`, что входная информация является корреляционной матрицей. В построенной сети каждый узел представляет один из 8 гидрохимических показателей набора данных `Tchim.class`, а ребра соответствует корреляционным связям между ними: зеленые ребра представляют положительные корреляции, а красные определяют отрицательный характер связи. Чем шире и насыщеннее цвет ребра, тем выше коэффициент корреляции.

Аргумент `minimum` задает пороговое значение, указывающее, что все ребра со значением ниже минимального становятся невидимыми (но учитываются при вычислении характеристик графа). Аналогично аргумент `maximum` устанавливает верхнюю границу для отображения ребер. Аргумент `cut` управляет масштабированием по насыщенности и ширине: ребра выше значения `cut` одинаково насыщены и масштабируются по ширине, тогда как ребра ниже значения `cut` имеют малую ширину и масштабируются по цвету.

Если используется аргумент `layout = "circular"`, то узлы размещаются в круге по часовой стрелке. Другой вариант с аргументом `layout = "spring"` перестраивает изображение, по возможности, размещая сильно коррелированные между собой узлы ближе друг к другу (как это используется в факторном анализе). Такая кластеризация узлов улучшает визуализацию и облегчает анализ компонентов сети. Для построения таких оптимизированных графиков используется взвешенная версия алгоритма Фрухтермана-Рейнгольда [Fruchterman, Reingold, 1991], а параметр `repulsion` определяет, насколько выраженной должна быть зависимость между длиной ребер на диаграмме и коррелированностью узлов. При построении следующего графика используем матрицу ранговых коэффициентов корреляции Спирмена:

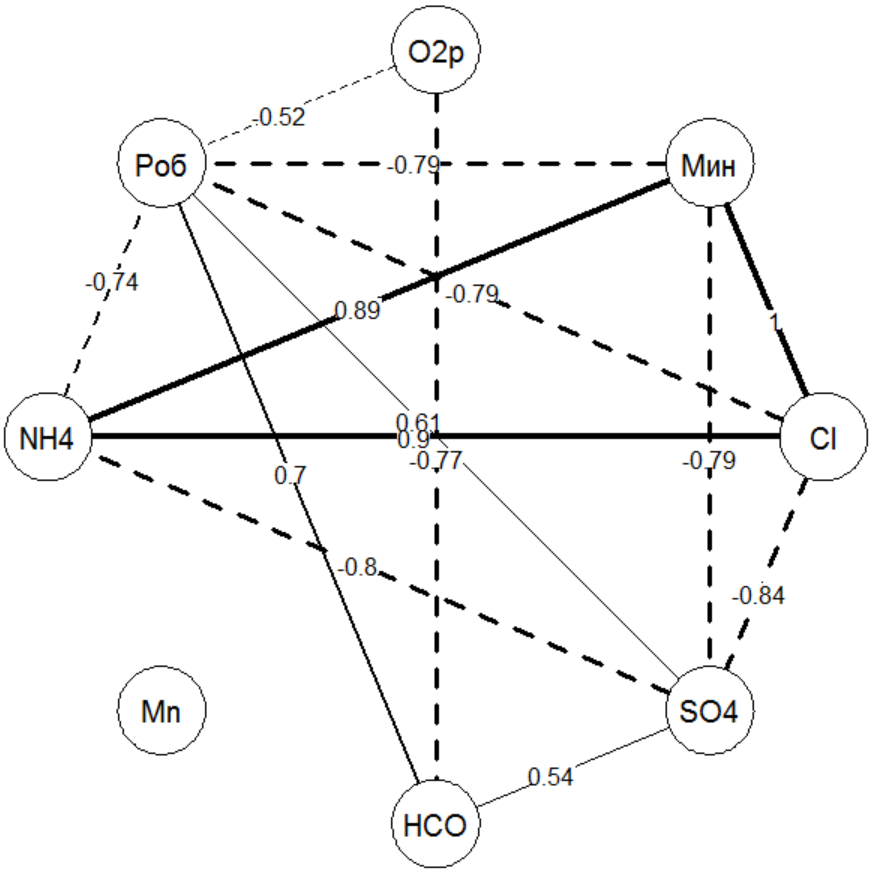
```
corGraph <- qgraph(Cor_S, layout = "spring", graph = "cor",
        maximum = 1, minimum = 0, repulsion = 0.8,
        edge.labels=T, edge.label.color= "black")
```



Функция `qgraph` включает много различных опций форматирования изображения сети и несколько десятков графических параметров, позволяющих управлять цветом, шрифтами или толщиной линий различных ее компонентов (см. документацию к пакету [qgraph, 2019]). Значения этих аргументов обычно выбираются по умолчанию, но их осознанное использование позволяет гибко подстроить вид графика с учетом конкретных требований. Например, чтобы вывести для каждого ребра значения коэффициентов корреляции можно использовать параметры `edge.labels` и `edge.label.color`.

Как упоминалось выше, можно удалить из сети все ребра, коэффициенты корреляции которых не являются статистически значимыми. Для этого устанавливается значение аргумента `minimum = "sig"` и, при необходимости, задается уровень значимости `alpha` или подключается процедура коррекции Бонферрони. Использование опции `theme = "gray"` дает возможность получить график сети, соответствующий стандартным требованиям, обычно предъявляемым к рукописям в научные журналы :

```
corGraph <- qgraph(Cor_P, layout = "circular", graph = "cor",
         cut = 0.5,  minimum = "sig",
         edge.labels=T, theme = 'gray',    esize =5,
         alpha=0.05, # уровень значимости
         bonf=F,     # использование поправки Бонферрони
         sampleSize=15    # Объем выборки
)
```



Отметим, что вместо матрицы коэффициентов корреляции может фигурировать любая адекватная поставленной задаче квадратная симметричная матрица величин, взвешивающих выраженность взаимодействий между каждой парой узлов. Если, предположим, нам надо оценить частоту совместной встречаемости двух биологических видов, то в качестве такой величины можно принять коэффициент сходства Брея-Кёртиса (он же индекс Ренконена, процентное подобие, коэффициент общности, индекс Штейнгауза, количественная мера сходства Чекановского и т.д.):

$$M_{xy} = 2\sum_{i=1}^{n} \min[x_i, y_i] / (\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i),$$

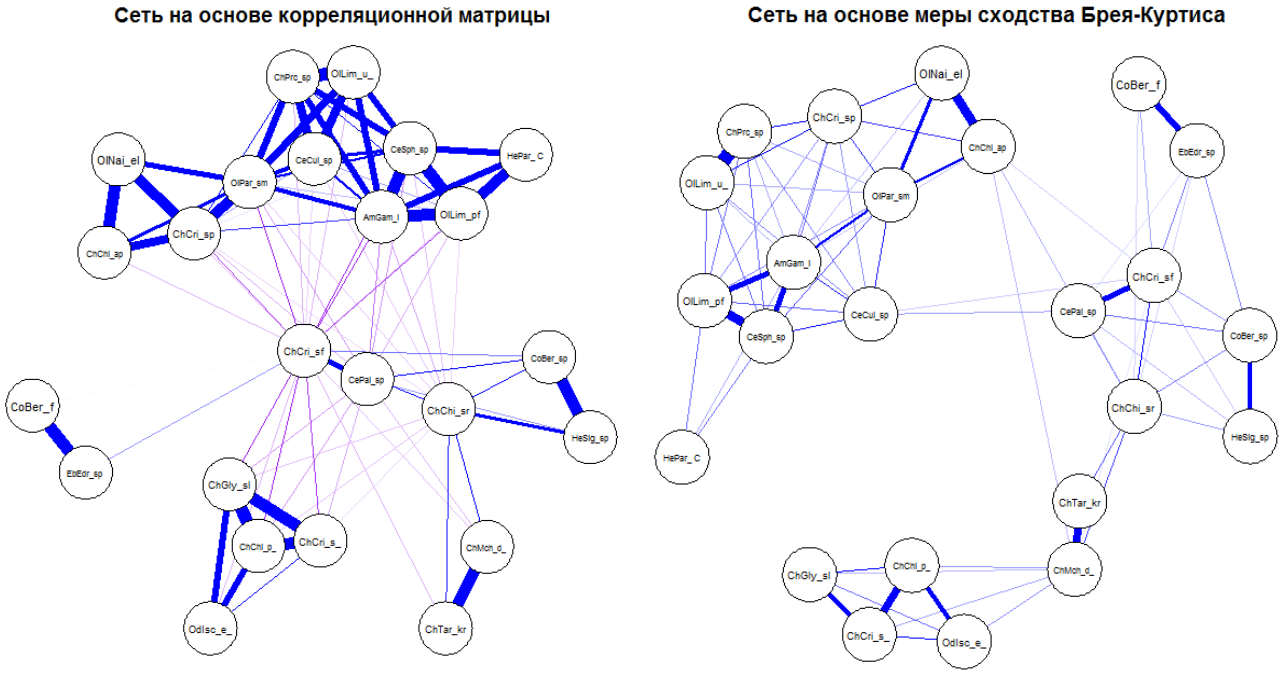
где $\{x_1, x_2, \ldots, x_n\}$ и $\{y_1, y_2, \ldots, y_n\}$ – удельные численности особей сравниваемых видов по результатам отбора n проб из экосистемы. Напомним, что в тех же обозначениях оценки коэффициентов корреляции и ковариации рассчитываются по формулам

$$r_{xy} = \mathrm{cov}(x, y) / \sigma_x \sigma_y, \qquad \mathrm{cov}(x, y) = \sum_{i=1}^{n} (x_i - m_x)(y_i - m_y),$$

где m_x, m_y – оценки средних и σ_x, σ_y – дисперсии векторов численностей видов.

Используем в качестве исходных данных таблицу оценок популяционной плотности 24 видов макрозообентоса в 15 точках экосистемы малых рек бассейна оз. Эльтон и сравним две сети, полученные на основе матрицы мер Брея-Кёртиса и корреляционной матрицы:

```
colnames(tab.ZB)
 [1] "AmGam_l"  "CeCul_sp" "CePal_sp" "CeSph_sp" "ChChi_ap" "ChChi_p_" "ChChi_sr" "ChCri_s_" "ChCri_sf"
[10] "ChCri_sp" "ChGly_sl" "ChMch_d_" "ChPrc_sp" "ChTar_kr" "CoBer_f"  "CoBer_sp" "EbEdr_sp" "HePar_ C"
[19] "HeSig_sp" "OdIsc_e_" "OlLim_pf" "OlLim_u_" "OlNai_el" "OlPar_sm"
qgraph(cor(tab.ZB), posCol = "blue", negCol = "purple",
       labels = colnames(tab.ZB), minimum = 0.2, layout = "spring")
title("Сеть на основе корреляционной матрицы", line = 2.5)
library(vegan)
Met_S <- 1 - vegdist(t(tab.ZB))
qgraph(Met_S, posCol = "blue", negCol = "purple",
       labels = colnames(tab.ZB), layout = "spring")
title("Сеть на основе меры сходства Брея-Куртиса", line = 2.5)
```



Смысл наших сомнений в целесообразности использования коэффициента корреляции Пирсона для оценки степени совместной встречаемости видов связан со структурой исходной таблицы, подавляющее большинство ячеек которой заполнено нулями (отсутствие вида в пробе является к тому же статистически неопределенной величиной). Тем самым нарушается предположение о нормальности анализируемых величин, принимаемое при оценке корреляции Пирсона. Мера Брея-Кёртиса априори не зависит от характера статистического распределения исходных данных и более объективно отражает процессы коинтеграции видов.

3. Группировка узлов

Группировка с помощью аргумента `group` позволяет указать, какие узлы относятся к одному и тому же типу объектов. При использовании группировки изображение сети модифицируется следующим образом:

- На диаграммах с круговой компоновкой (`layout = "circular"`) каждая группа образует свой круг меньшего размера;
- Узлы одного типа окрашены в один и тот же цвет, принимаемый либо на основе стандартной палитры, либо определенный с помощью параметра `color`;
- Наименования групп могут быть представлены в легенде.

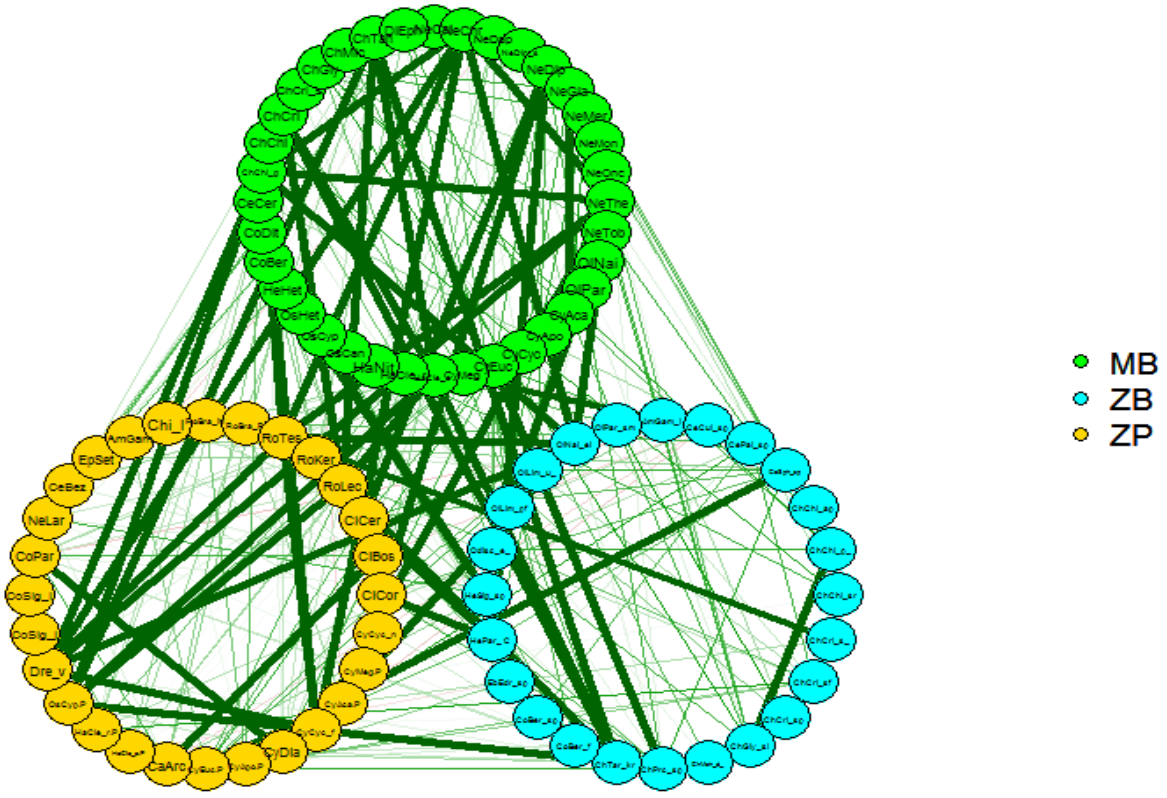
Группировка задается при помощи фактора (вектора символьных обозначений групп), либо списком с номерами узлов, принадлежащих каждой группе.

Объединим таблицы с наблюдениями трех сообществ 88 видов гидробионтов и обозначим каждую группу следующим образом: `MB` - мейобентос, `ZB` - зообентос и `ZP` – зоопланктон:

```
tab <- cbind(tab.ZB,tab.ZP)
tab <- as.data.frame(cbind(tab, tab.MB))
gF <- factor(c(rep("ZB", ncol(tab.ZB)),
      rep("ZP", ncol(tab.ZP)), rep("MB", ncol(tab.MB))))
```

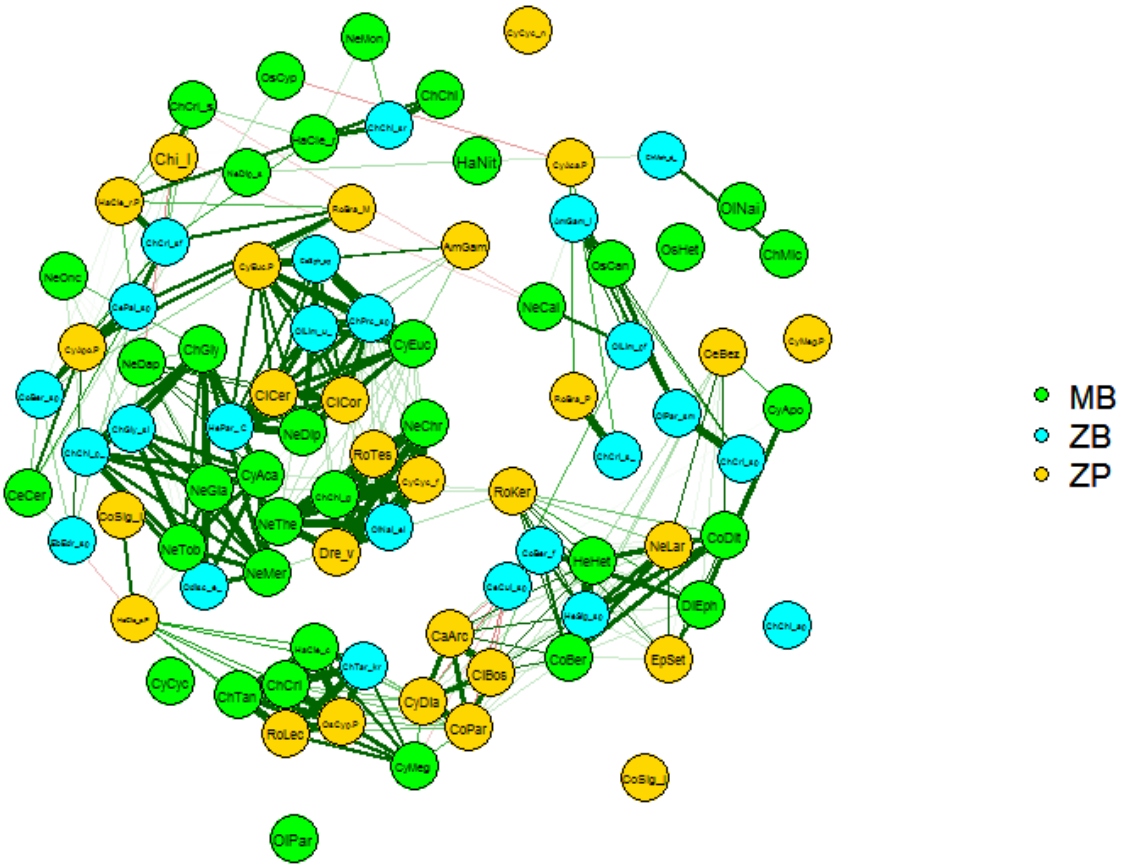
Поскольку обилие таксонов оценивалось по шестибалльной шкале, предположение о нормальности очевидно нарушается. В качестве возможного решения вычислим коэффициенты полихорической корреляции. Для этого можно использовать функцию `cor_auto()` из пакета `qgraph`, которая автоматически определяет, являются ли данные порядковыми, и запускает функцию `lavCor()` из пакета `lavaan` для вычисления Пирсоновых, полихорических или полисериальных корреляций там, где это необходимо:

```
AssocCors <- cor_auto(tab)
AsocGraph <- qgraph(AssocCors, graph = "cor", minimum = 0.75,
      color=c("green", "cyan", "gold"), labels = colnames(tab),
      cut = 0.99,  groups = gF,legend = TRUE,  esize = 5)
```



Полученный объект `AsocGraph` можно использовать для той же функции `qgraph()`, изменив при необходимости некоторые ключевые аргументы:

```
qgraph(AsocGraph, layout = "spring",cut = 0.85, minimum = 0.75)
```



4. Частная корреляционная сеть

Как отмечалось выше, при интерпретации взаимозависимостей между показателями часто наблюдается эффект "ложной корреляции": если получен значимый коэффициент корреляции какой-то случайной величины с другой, то это может быть всего лишь отражением того факта, что обе они коррелируют с некоторой третьей величиной или с совокупностью величин (измеренных или латентных). Ложные корреляции порождают многочисленные факты нарушения взаимной обусловленности связей, например, два вида *Chironomus plumosus* (ChChi_p) и *Cricotopus* sp. (ChCri_sp) имеют значимую обратную функциональную зависимость с третьим видом *Palpomyia* sp. (CePal_sp) при коэффициенте ранговой корреляции Спирмена r = -0.9712, тогда как между ними самими также обнаруживается отрицательная корреляция r = -0.9276, что логически трудно объяснимо.

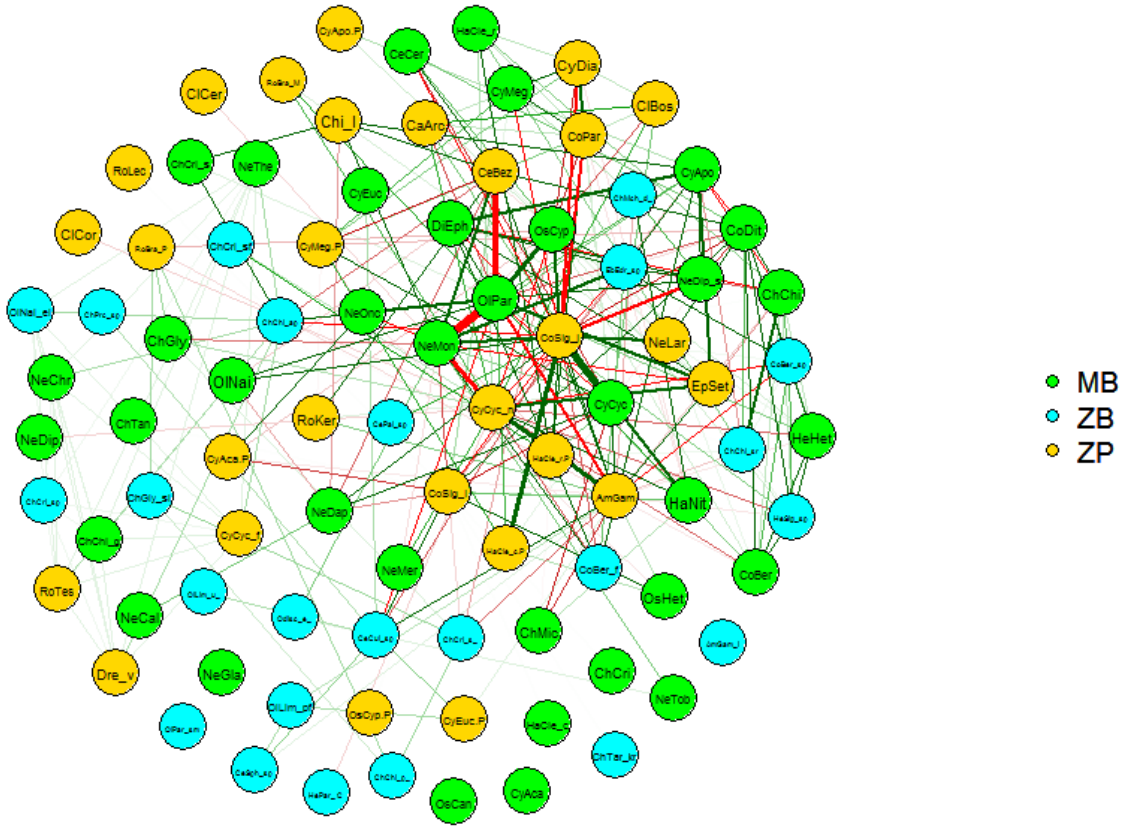
Эта ситуация приводит к необходимости анализа так называемых *частных корреляций*, т.е. условных корреляций между двумя величинами при *фиксированных* значениях остальных величин. Коэффициенты частной корреляции моделируют "чистые" эффекты парного взаимодействия между двумя конкретными узлами сети после элиминации влияния всех других узлов. Однако узлы, не связанные с анализируемой парой, считаются только условно независимыми, поскольку обладают некоторой остаточной корреляцией. Таким образом, веса ребер частной корреляционной сети можно рассматривать как некоторую прогностическую оценку коэффициентов выраженности связей между парами узлов, что аналогично оцениванию коэффициентов множественной регрессии.

Вычислительные аспекты частной (или парциальной) корреляционной сети непосредственно связаны с обратной корреляционной (или ковариационной) матрицей **P**. Для нормально распределенных непрерывных величин с индексами i и j коэффициенты частной корреляции ω_{ij} можно получить по формуле $\omega_{ij} = -p_{ij} / \sqrt{p_{ii} p_{jj}}$. Отсюда требование к исходной корреляционной матрице – она должна быть положительно определенной.

Симметричная квадратная матрица положительно определена, если все ее собственные числа больше нуля. Наличие отрицательных собственных чисел корреляционной матрицы обычно свидетельствует о ее неопределенности и вырожденности. Действительно, большое число редких видов встретилось только в одной-двух пробах из 15 и (при случайном совпадении их встреч) коэффициенты корреляции между ними в таблице `AssocCors` близки к 1, хотя здравого статистического смысла при такой оценке силы связи немного. Другим несомненным условием неотрицательности собственных чисел корреляционной матрицы является превышение объема выборки n над числом переменных p, т.е. $n > p$.

Если условие положительной определенности не выполняется, то одним (но не лучшим) выходом является найти ближайшую положительно определенную матрицу, например, методом аппроксимации по [Higham, 2002] с использованием функции `nearPD()`, а уже потом использовать ее для расчета частных коэффициентов корреляции.

```
# подсчет числа отрицательных собственных чисел
sum(eigen(AssocCors)$values < 0)
[1] 50
library(Matrix)
AssPDCors <- nearPD(AssocCors, corr = TRUE, maxit = 250)
sum (eigen(AssPDCors$mat)$values < 0)
[1] 0
AssPDCors <- AssPDCors$mat
AsocGraph <- qgraph(AssPDCors, graph = "pcor", cut = 0.1, minimum = 0.07,
      color=c("green", "cyan", "gold"), labels = colnames(tab), esize = 5,
      layout = "spring",repulsion = 0.7, groups = gF, legend = TRUE)
```



Вычисленные коэффициенты частной корреляции значительно ниже, чем обычные коэффициенты корреляции, и только 11 из них превышают значение 0.1. На приведенном выше графике показаны ребра выше заданного минимума `minimum = 0.07`, однако в пакете `qgraph` предусматривается возможность автоматического вычисления порогового значения с использованием аргумента `threshold`. При этом (как и при использовании аргумента `minimum = "sig"`) выполняются различные статистические тесты, оценивающие значимость частных коэффициентов корреляции с большими дополнительными возможностями. Например, мы можем удалить все незначимые ребра, используя при этом коррекцию Хольма, следующим образом:

```
AsocGraph <- qgraph(AssPDCors, graph = "pcor", cut = 0.1, threshold = "holm",
      color=c("green", "cyan", "gold"), labels = colnames(tab),
      layout = "spring",repulsion = 0.7,  sampleSize=15,
      groups = gF,legend = TRUE,  esize = 5)
Предупреждения:
1: В sqrt(n - 2) : созданы NaN
2: В psych::corr.p(input, n = nadj, adjust = threshold, alpha = max(alpha)) :
  Number of subjects must be greater than 3 to find confidence intervals.
3: В sqrt(n - 3) : созданы NaN
4: В qgraph(AssPDCors, graph = "pcor", cut = 0.1, threshold = "holm",  :
  Non-finite weights are omitted
```

В связи с плохой обусловленностью исходной матрицы и малым объемом выборки выполнить процедуру не удалось.

5. Поиск оптимального графа сети

Сети, заданные корреляционными матрицами, как правило, полностью насыщены, хотя некоторые (частичные) коэффициенты корреляции настолько малы, что практически невидимы. Цель выбора модели состоит в том, чтобы в полностью насыщенной сети определить, какие связи равны нулю, поскольку эти значения указывают на отношения условной независимости узлов. Выше мы использовали простые способы сделать это: устанавливали пороговые значения наобум, либо оценивали статистическую значимость коэффициентов корреляции.

Удаление незначимых ребер с использованием пороговых значений – довольно грубый способ выбора модели сети, поскольку оцениваются локальные коэффициенты корреляции и не рассматривается, как изменяются значения другие ребер при исключении части из них. Процедура построения оптимальной сети ставит задачу иначе: какой граф из множества всех возможных моделей сети адекватно объясняет исходные данные, будучи наиболее простым (т.е. имеющим наименьшее число ребер). Для этого необходимо оценить все возможные модели и сравнить их между собой по некоторому информационному критерию, который, грубо говоря, взвешивает потери информации в результате разрежения сети. Обычно для гауссовых графических моделей используется усовершенствованный байесовский критерий EBIC (Extended Bayesian Information criterium – Foygel, Drton, 2010).

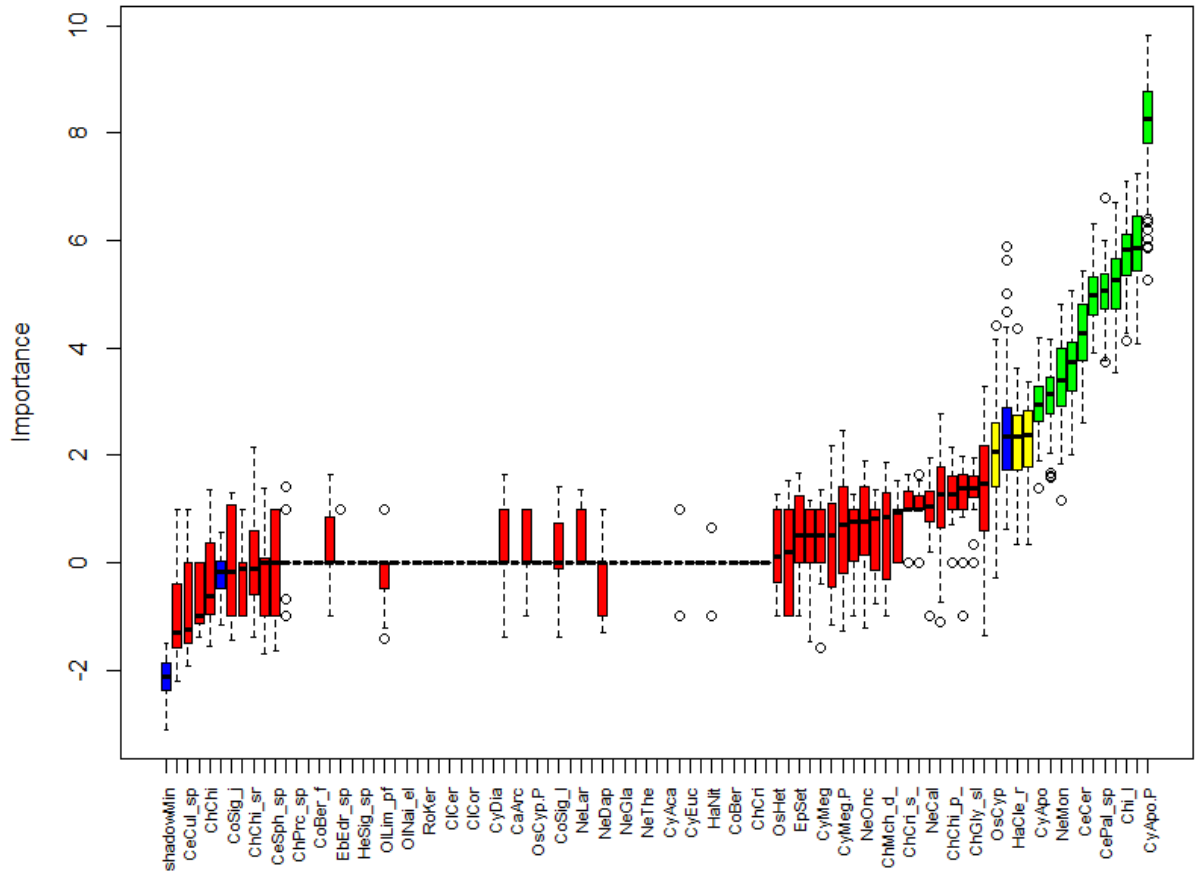
Поскольку число возможных моделей равно $2^{m\,(m-1)/2}$, где m – количество узлов в сети, то уже для случая 25 узлов сравнить все возможные модели становится нереальным и выполняется направленный пошаговый поиск в пространстве моделей. Можно начать селекцию снизу с модели полностью независимых узлов, и постепенно переходить выше к более сложным моделям, или стартовать с насыщенной модели и, постепенно обрезая "лишние" связи, рассматривать более простые разреженные графы: разные варианты перебора сети задаются аргументами функции `ggmModSelect()`. Однако время поиска для 88 узлов оказалось столь велико, что нам не удалось дождаться какого либо результата:

```
cor.opt <- ggmModSelect(AssPDCors, n=15)$graph
```

Возможным выходом из создавшейся ситуации является снижение размерности исходной матрицы наблюдений с использованием внешних процедур. Например, можно удалить столбцы с редкими видами, обнаруженными в 1-2-х пробах или объединить столбцы наблюдений с экологически близкими таксонами. Существуют также различные статистические методы селекции информативно значимых переменных, в том числе и для сетевых моделей (о чем речь пойдет в последующих разделах).

Однако, с учетом характера изучаемых нами проблем, осуществим отбор подмножества видов, изменчивость обилия которых статистически значима относительно градиента солености воды. Фактор `Min.class` таблицы `Tchim.class` задает следующие классы минерализации (Мин): 1 – от 25 до 35 г/л; 2 – от 10 до 25 г/л; 3 – менее 10 г/л. Для оценки статистической значимости используем алгоритм "Борута", основанный на многократном повторении процедуры классификации "случайный лес" [Шитиков, Мастицкий, 2017]. Скрипт, представленный ниже, осуществляет оценку "важности" (*importance*) каждого из 88 исходных таксонов и дает заключение о его статистической значимости. Незначимые переменные на представленном графике показаны красным цветом, статистически значимые – зеленым цветом, а находящие в "пограничном" состоянии – желтым.

```
library(Boruta)
Y <- Tchim.class[,9]
mod.Boruta <- Boruta(Y ~ ., data = tab,
            doTrace = 2, ntree = 500)
plot(mod.Boruta, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(mod.Boruta$ImpHistory),
   function(i)
    mod.Boruta$ImpHistory[is.finite(mod.Boruta$ImpHistory[,i]),i])
names(lz) <- colnames(mod.Boruta$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
     at = 1:ncol(mod.Boruta$ImpHistory), cex.axis = 0.7)
```

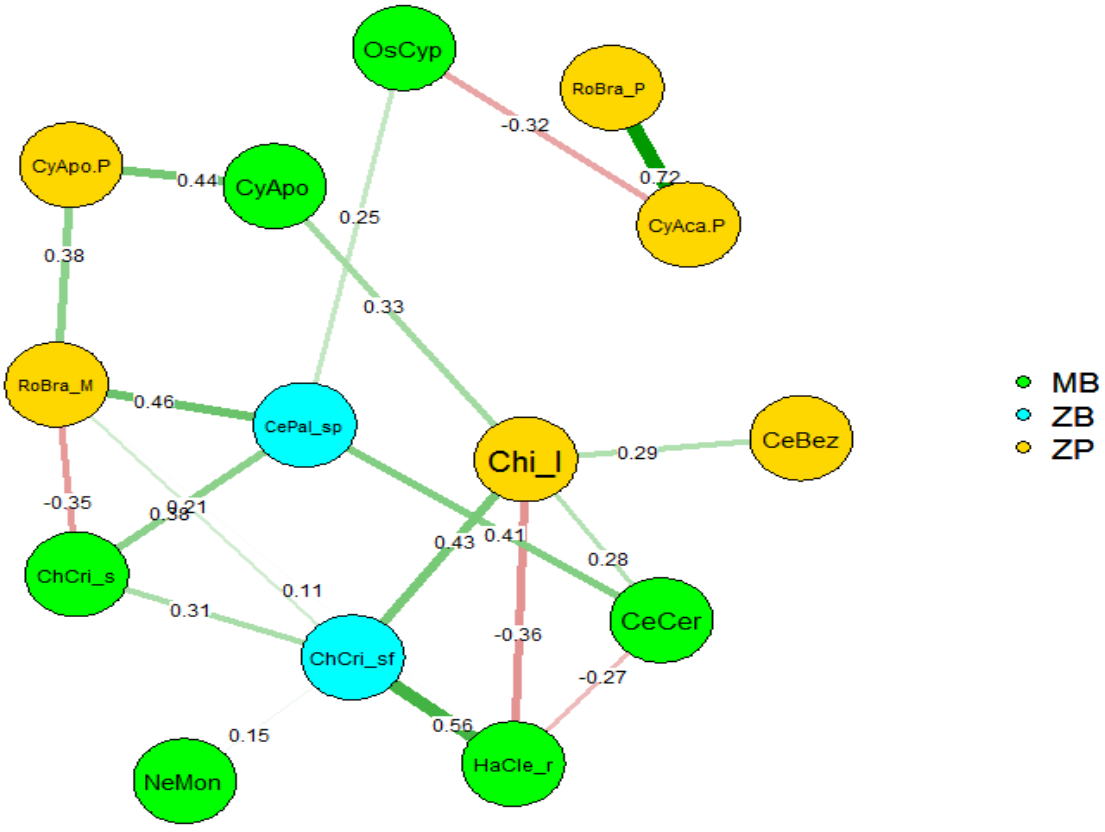


Как следует из графика результатов "Борута", статистически значимо связаны с соленостью воды только 14 видов из 88. Сформируем новые исходные матрицы с сокращенным набором признаков, который включает только статистически значимые виды:

```
getConfirmedFormula(mod.Boruta)
ind <- attStats(mod.Boruta)$decision != "Rejected"
mod.tab <- tab[,ind]
mod.gF   <- gF[ind]
mod.cor <- cor(mod.tab)
```

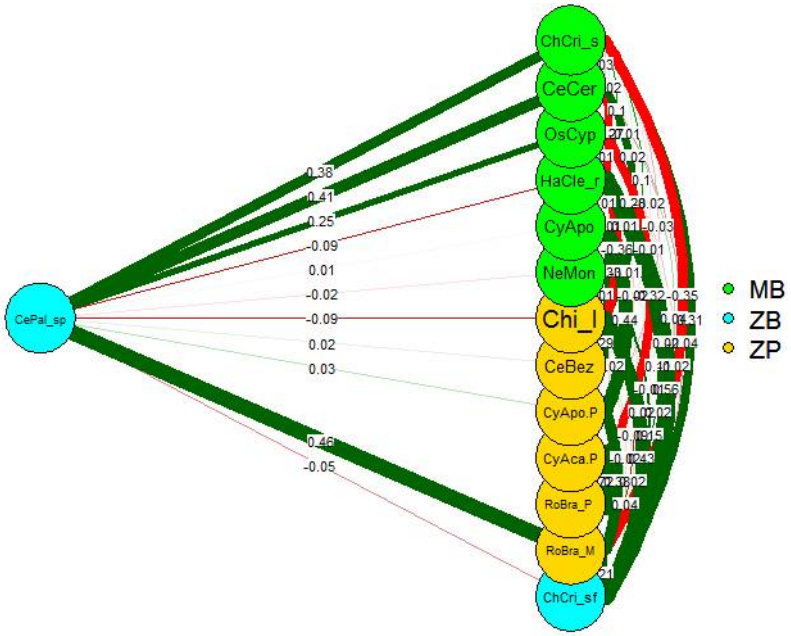
Для матриц с сокращенным списком видов с использованием функции `ggmModSelect()` получим оптимальную корреляционную сеть:

```
mod.cor.opt <- ggmModSelect(mod.cor, n=15)$graph
diag(mod.cor.opt) <- 1
AsocGraph <- qgraph(mod.cor.opt, graph = "cor", minimum = 0.1,layout = "spring",
      color=c("green", "cyan", "gold"), labels = colnames(mod.tab),
      cut = 0,  groups = mod.gF,legend = TRUE,  esize = 10, repulsion = 0.8,
      edge.labels=T, edge.label.color= "black", edge.label.cex=0.7)
```



Иногда исследователя интересуют связи одного из узлов со всеми остальными, что бывает трудно увидеть в графе Фрухтермана-Рейнгольда, особенно когда связи с этим одним узлом слабые. Для этого может быть использована функция `flow()`, которая размещает узлы таким образом, чтобы корреляционные связи одного конкретного узла были хорошо видны:

```
flow(AsocGraph, "CePal_sp")
```



6. Связь пакета qpraph с методами многомерного анализа

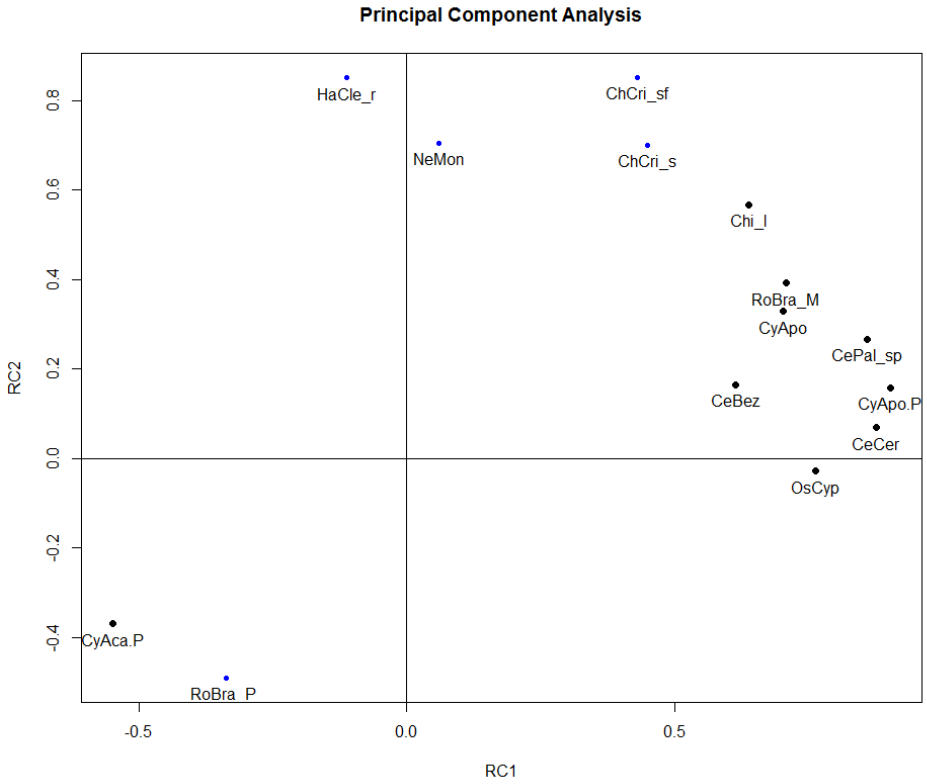
Рассмотрим теперь методы решения двух возможных задач:

- Можно ли разместить узлы графа на диаграмме таким образом, чтобы их координаты соответствовали результатам предварительно проведенной ординации https://stok1946.blogspot.com/2019/03/blog-post.html ?
- Как можно построить граф, отражающий связи не только между узлами, но и с некоторыми группообразующими латентными переменными, полученными в ходе факторного анализа?

Если задать аргумент `layout = "spring"`, то используется алгоритм Фрухтермана-Рейнгольда, формирующий граф "под действием векторов сил" (*force-directed*), который сродни созданию физической системы из шаров, соединенных упругими нитями. Однако такое расположение узлов не всегда легко поддается содержательному анализу. Существуют и другие подходы к построению графов сетей, которые позволяют внести в схему позиционирования узлов содержательно интерпретируемый смысл. Одним из таких является использование методов многомерного статистического анализа: главных компонент PCA, многомерного шкалирования MDS и проч. [Jones et al., 2018].

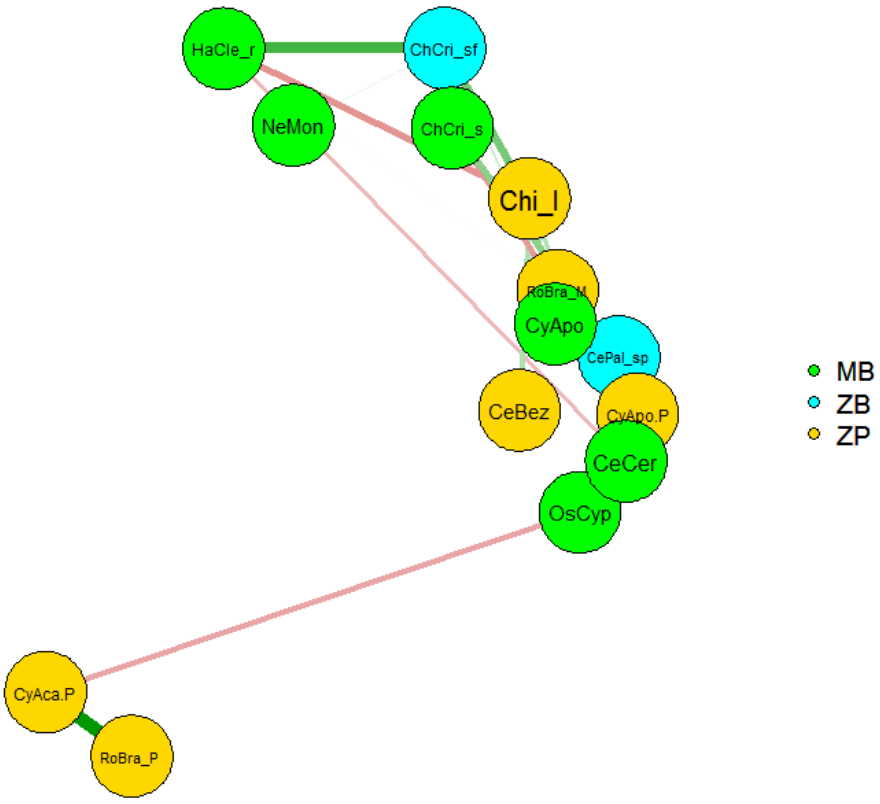
Выполним анализ главных компонент корреляционной матрицы `mod.cor` с использованием функции `principal()` из пакета `psych`:

```
library(psych)
PCA_adult <- principal(mod.cor, nfactors = 2)
plot(PCA_adult, label=colnames(mod.tab))
```



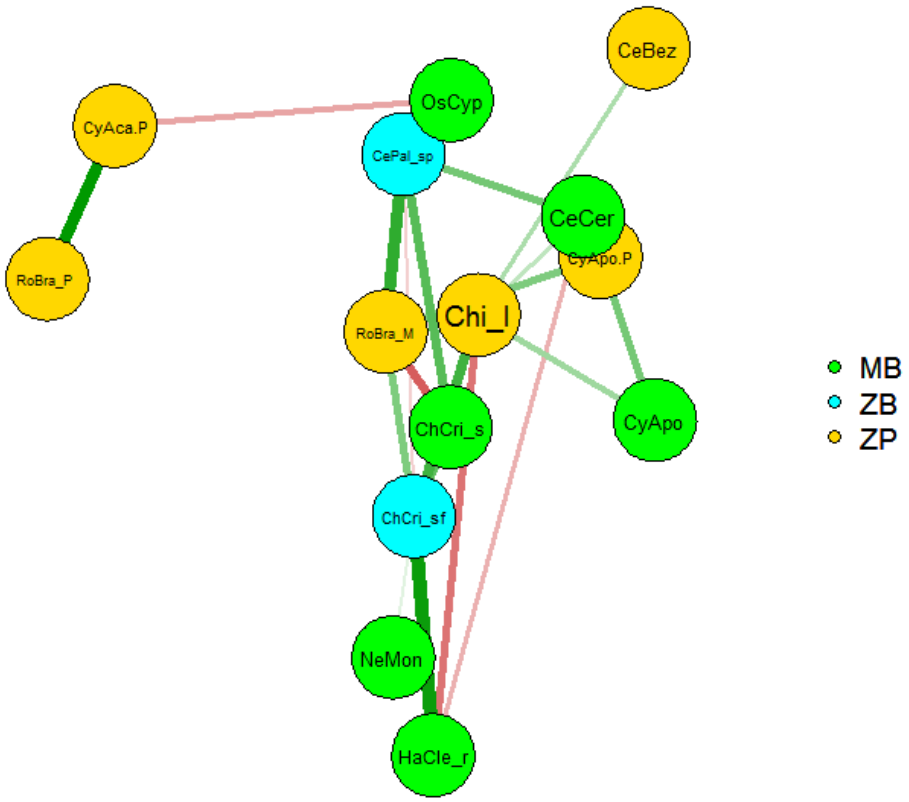
Если задать аргумент `layout = PCA_adult$loadings`, то координаты узлов формируемой сети будут находиться в соответствии со значениями нагрузок `loadings` вычисленных в ходе PCA-анализа:

```
AsocGraph <- qgraph(mod.cor.opt, graph = "cor", minimum = 0.1,
    layout = PCA_adult$loadings, color=c("green", "cyan", "gold"),
    labels = colnames(mod.tab), cut = 0,  groups = mod.gF, legend = TRUE,
    esize = 10, repulsion = 0.8)
```



Аналогичным образом мы можем использовать результаты процедуры многомерного неметрического шкалирования на основе матрицы расстояний Брея-Кёртиса:

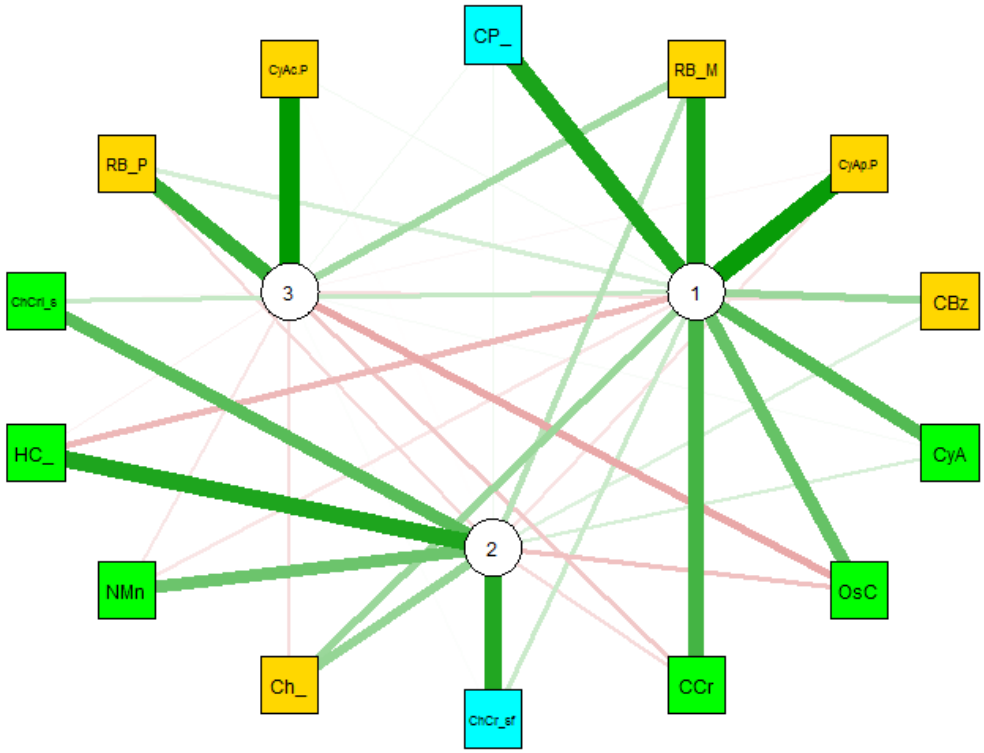
```
library(vegan)
mod.mds <- metaMDS(t(mod.tab), trace = FALSE)
AsocGraph <- qgraph(mod.cor.opt,  minimum = 0.1, layout = mod.mds$points,
      color=c("green", "cyan", "gold"), labels = colnames(mod.tab),
      cut = 0,  groups = mod.gF,legend = TRUE,  esize = 10, repulsion = 0.8)
```



Результатом факторного анализа (или PCA) является матрица факторных нагрузок `loadings`, которая содержит оценки величины связи каждой исходной переменной элемента с набором выделенных латентных факторов (главных компонент). Типичным способом интерпретации такой матрицы является выделение жирным шрифтом факторных нагрузок, которые выше или ниже заданного порога. При таком подходе небольшие, но интересные нагрузки могут быть легко пропущены. С использованием пакета `qgraph` легко можно визуализировать (аналогично корреляционным матрицам) матрицы факторной нагрузки, используя их величины в качестве весов ребер сети.

В качестве функции, выполняющей факторный анализ, авторы пакета предлагают использовать `factanal()`, аргументами которой должны быть корреляционная матрица, число формируемых факторов `factors` (равное, например, числу групп) и метод желаемого вращения системы факторных координат `rotation`. Матрица нагрузок в качестве аргумента передается функции `qgraph.loadings()` для формирования графика:

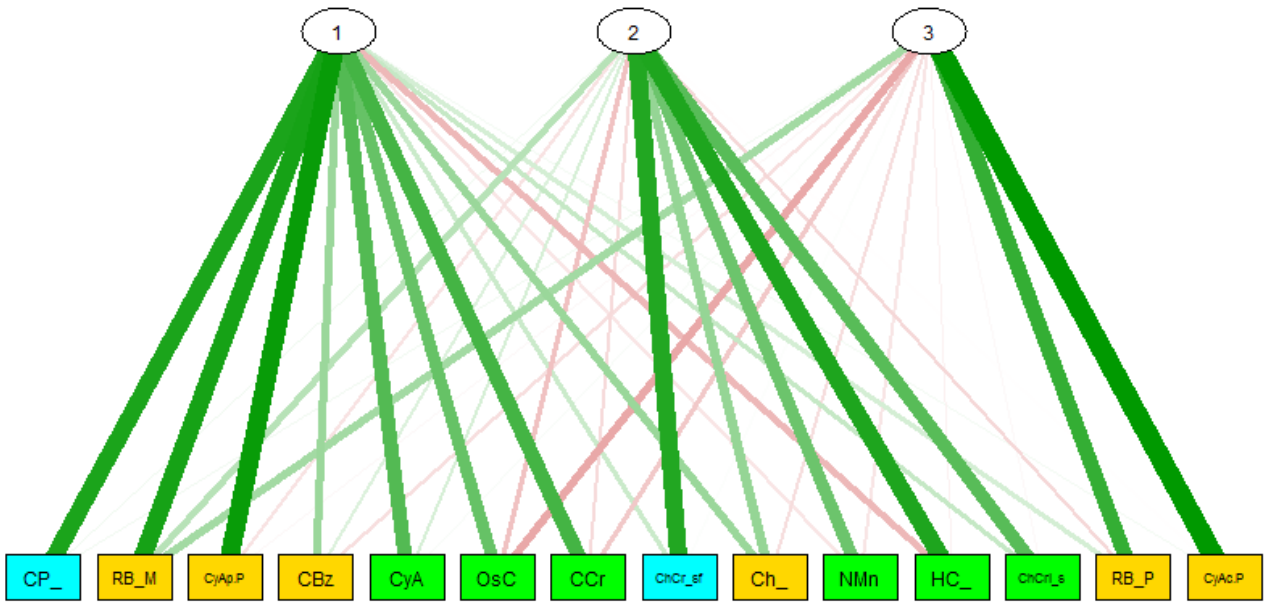
```
FacRes <- factanal(mod.tab,factors=3,rotation="promax",scores="regression")
FacResLoad <- loadings(FacRes)
qgraph.loadings(FacResLoad,groups=mod.gF,rotation="promax",
    color=c("green", "cyan", "gold"))
```



На представленном графике исходные переменные располагаются по внешней окружности вблизи того фактора, на который они оказывают наибольшее влияние (т.е. в отношение которого они имеют наибольшее значение нагрузки)

Другой (более привычной) формой того же графика является расположение узлов в виде дерева `layout="tree"`:

```
qgraph.loadings(FacResLoad,groups=mod.gF,rotation="promax",
      color=c("green", "cyan", "gold"),  layout="tree")
```



7. Построение модели сети с использованием алгоритмов регуляризации

Как мы убедились выше, оценка значений неизвестной положительно определенной ковариационной матрицы $\mathbf{\Sigma}$ для таблицы реализаций $\mathbf{X}n\times p$ из p-мерного гауссова распределения при $n \ll p$ является сложной задачей, поскольку обычная процедура MLE метода максимального правдоподобия становится не выполнимой. "Графическое лассо" (*graphical lasso* – Friedman et al, 2007) является алгоритмом оценки ковариационной матрицы Σ с использованием регуляризации и в предположении, что обратная матрица $\boldsymbol{\theta} = \mathbf{\Sigma}^{-1}$ является разреженной. Если в такой матрице, называемой прецизионной (*precision*), элементы $\theta_{jk} = 0$, то это означает, что соответствующие переменные x_j и x_k условно независимы (с учетом взаимодействий с остальными переменными). С использованием метода графического лассо ищется минимум отрицательного логарифма следующей l_1-регуляризируемой функции

$$\underset{\theta \succ 0}{\text{minimize}}\ f(\boldsymbol{\theta}) := -\log\det(\boldsymbol{\theta}) + \text{tr}(\mathbf{S}\,\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}\|_1 \ ,$$

где $\mathbf{S}$ – выборочная ковариационная матрица, а $\|\boldsymbol{\theta}\|_1$ обозначает сумму абсолютных значений $\boldsymbol{\theta}$. Таким образом, представленным методом регуляризации находится оценка максимального правдоподобия матрицы $\mathbf{\Sigma}$ с учетом штрафа за сумму абсолютных значений обратной ковариационной матрицы, где размер штрафа задается параметром настройки λ. В результате строится графическая модель сети, которая все еще достаточно хорошо объясняет данные, но многие коэффициенты связей обращаются в нуль.

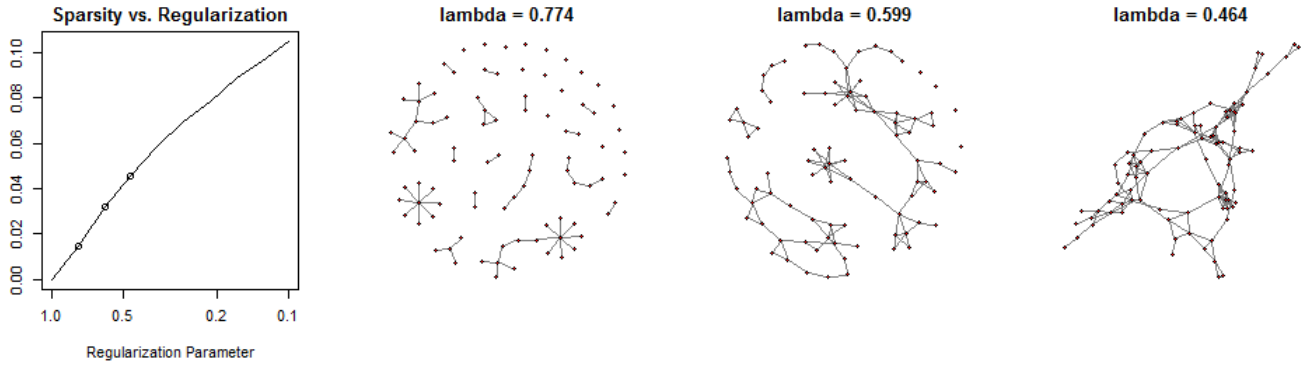
Пакет `qgraph` содержит функцию `EBICglasso()`, которая реализует алгоритм графического лассо, где параметр λ настраивается с использованием усовершенствованного байесовского информационного критерия (EBIC). Однако эта процедура требует большого числа наблюдений n, поэтому, к сожалению, в нашем случае получить непустую сеть методом `"glasso"` не удалось, несмотря на широкий диапазон изменения настроечных параметров:

```
AsoCGraph <- qgraph(AssPDCors, graph = "glasso", cut = 0.4, layout = "spring",
      repulsion = 0.7, maximum = 1, minimum = 0, lambda.min.ratio = 4.0,
      color=c("green", "cyan", "gold"), labels = colnames(tab), sampleSize=15,
      groups = gF,legend = TRUE,  esize = 5)
An empty network was selected to be the best fitting network.
Possibly set 'lambda.min.ratio' higher to search more sparse networks.
You can also change the 'gamma' parameter to improve sensitivity (at the cost of
specificity).
```

Для подбора оптимальных неориентированные графов по данным высокой размерности разработаны в R специализированные пакеты, такие как `huge`, который содержит простые в использовании функции, эффективно выполняющие процедуры регуляризации. В частности, функция `huge()` поддерживает два метода оптимизации сети: (i) алгоритм поиска ближайших узлов с построением штрафной регрессии [Meinshausen, Buhlmann, 2006] и (ii) графический алгоритм лассо [Friedman et al., 2007].

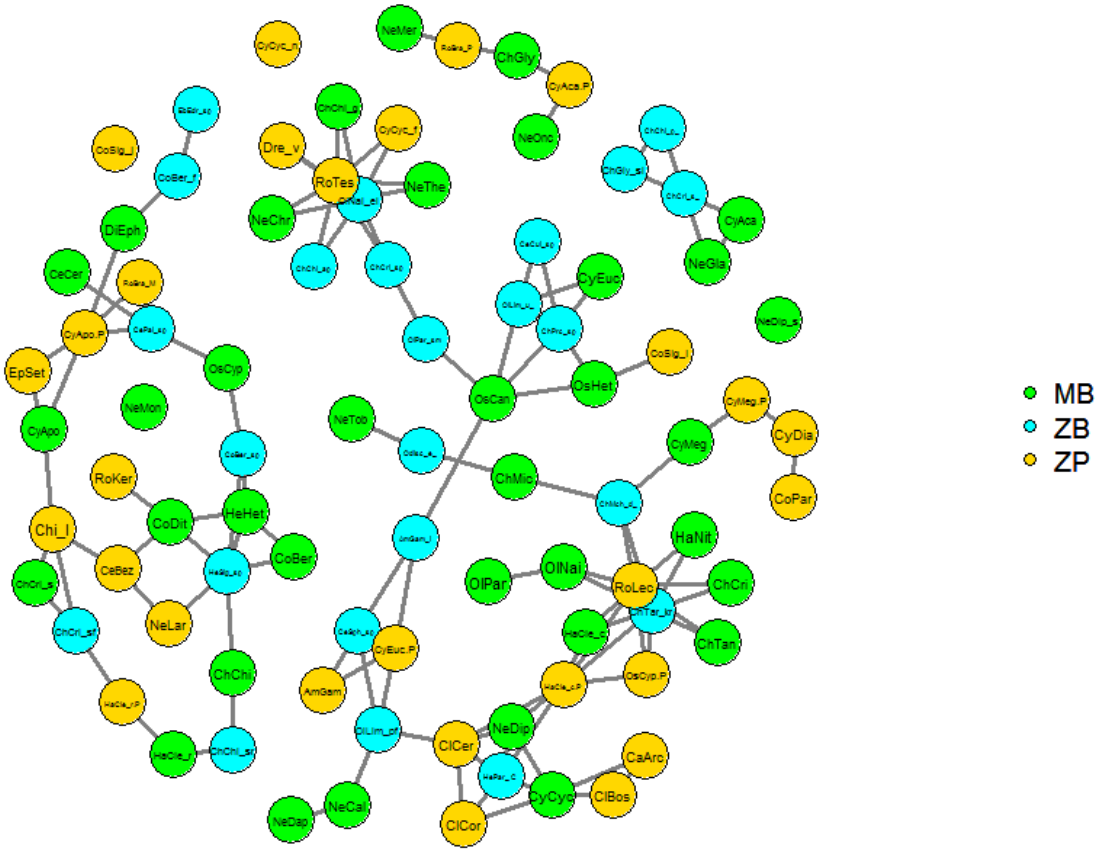
```
library(huge)
# по умолчанию используется алгоритм Meinshausen & Buhlmann
out.mb = huge(as.matrix(tab))
Model: Meinshausen & Buhlmann graph estimation (mb)
Input: The Data Matrix
Path length: 10
Graph dimension: 88
Sparsity level: 0 -----> 0.1047544
plot(out.mb)
```

По умолчанию формируется 10 вариантов графов сети для последовательности значений параметра регуляризации. При увеличении λ уменьшается разреженность сети (т.е. увеличивается ее связность). Три варианта сети представлены на графике:



Конфигурацию одного из вариантов (3-го) графа сети можно представить с использованием функций пакета `qgraph`:

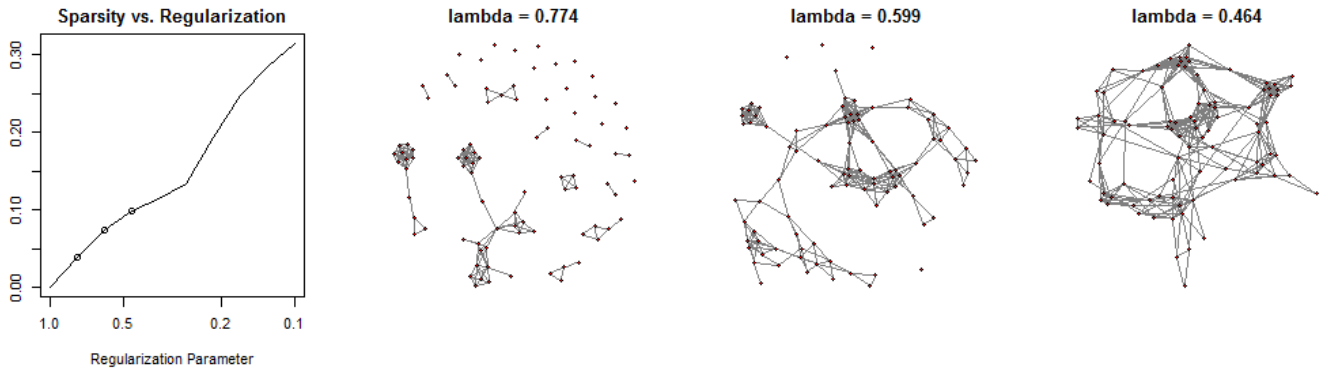
```
AsocGraph <- qgraph(out.mb$path[[3]], layout = "spring", repulsion = 0.9,
      color=c("green", "cyan", "gold"), labels = colnames(tab),
      groups = gF,legend = TRUE,  esize = 3)
```



Используем теперь алгоритм графического лассо:

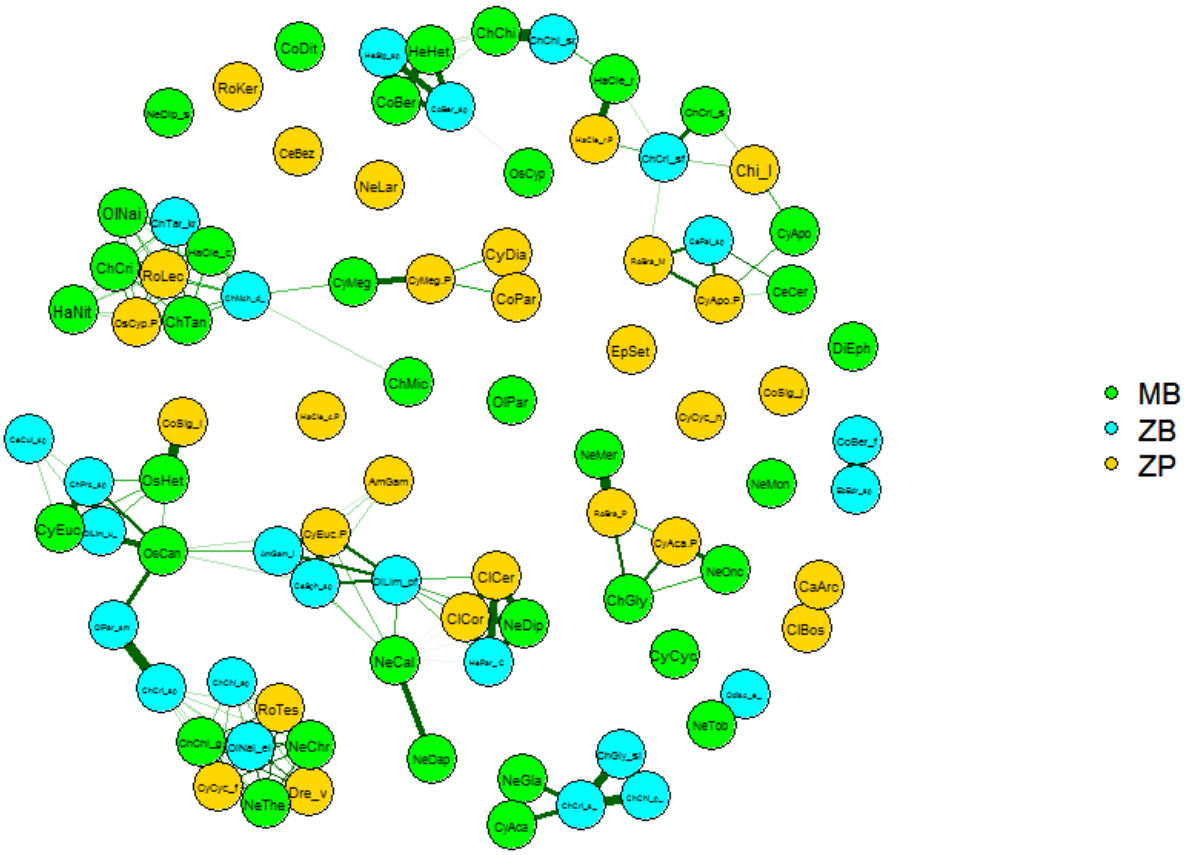
```
out.gb = huge(as.matrix(tab), method = "glasso")
Model: graphical lasso (glasso)
Input: The Data Matrix
Path length: 10
Graph dimension: 88
Sparsity level: 0 -----> 0.3145246
```

```
plot(out.gb)
```



Для визуализации графа, полученного методом Meinshausen, Buhlmann, мы использовали квадратную матрицу `out.mb$path[[3]]` связей (1 – есть связь, 0 – нет). Однако на основе прецизионной матрицы легко получить матрицу частных корреляций и использовать ее для количественной оценки выраженности статистической связи между видами:

```
mpc <-  wi2net(out.gb$icov[[3]])
AsocGraph <- qgraph(mpc,  minimum = 0.05,layout = "spring",
      color=c("green", "cyan", "gold"), labels = colnames(tab),
      groups = gF,legend = TRUE,  esize = 10)
```



Исходя из изложенного, в "лассо" задача поиска оптимальной сети, попросту говоря, подменяется задачей поиска оптимального значения параметра λ. Существуют алгоритмы поиска этого оптимума на формальных информационных критериев, но они работоспособны лишь при достаточно большом объеме выборочных данных.

Список литературных источников

Epskamp S., Cramer A., Waldorp L., Schmittmann V., Borsboom D. `qgraph`: Network visualizations of relationships in psychometric data.// Journal of Statistical Software. 2012. V. 48. P. 1-18.

Epskamp S. et al. `qgraph`: Graph Plotting Methods, Psychometric Data Visualization and Graphical Model Estimation. Version 1.6.4. 2019. Package R Development Core Team. https://cran.r-project.org/web/packages/qgraph/index.html

Epskamp S. Network Model Selection Using `qgraph` 1.3. The Psychosystems project. 2014. http://psychosystems.org/network-model-selection-using-qgraph-1-3-10/

Jones P.J., Mair P., McNally R.J. Visualizing Psychological Networks: A Tutorial in R. // Front. Psychol. 2018. V. 9. P. 1742. doi: 10.3389/fpsyg.2018.01742

Шитиков В.К., Мастицкий С.Э. Классификация, регрессия и другие алгоритмы Data Mining с использованием R. Электронная книга, 2017. 351 с. [Электронный ресурс] https://stok1946.blogspot.com (Дата обращения: 24.07.2018)

Friedman J., Hastie T., Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. // Biostatistics. 2007. V. 9. P. 432–441.

Meinshausen N., Buhlmann P. High dimensional graphs and variable selection with the lasso. // Annals of Statistics. 2006. V. 34(3). P. 1436–1462.