Ionflow: Ionomics data network and enrichment analysis

Wanchang Lin

01-12-2020

Contents

This document explains how to performs ionomics data analysis including gene network and enrichment analysis.

Data preparation

To explore the pipeline, we'll use the ionomics data set:

```
ion_data <- read.table("../test-data/iondata.tsv", header = T, sep = "\t")
dim(ion_data)
#> [1] 9999   16
```

Ten random lines are shown as:

```
sample_n(ion_data, 10)
```

Table 1: Samples of raw data

Knockout	Batch_ID	Ca	Cd	Co	Cu	Fe	K	Mg	Mn	Mo	Na	Ni	P	S	Zn
YLR154C	21	36.18	0.87	0.19	2.02	9.57	2809.04	603.63	1.27	1.14	186.98	1.25	4370.11	437.75	15.95
YER046W-A	14	50.76	1.02	0.17	2.06	10.35	3530.06	739.59	1.59	1.78	218.58	1.68	5002.07	584.78	19.98
YHR182W	19	32.18	0.95	0.12	1.35	4.23	2128.65	569.98	1.08	1.05	135.56	0.86	3888.56	386.84	14.16
YGR170W	16	103.92	1.20	0.17	1.87	9.82	3190.42	1150.56	1.56	2.58	248.90	1.50	6190.48	938.89	23.63
YDL227C	3	32.30	0.80	0.17	1.99	10.56	2808.39	493.74	1.22	1.07	222.72	1.24	3702.27	402.10	17.23
YOR209C	26	37.62	1.21	0.15	1.74	6.98	3527.69	756.27	1.47	1.20	209.94	1.18	5385.19	500.28	17.87
YER062C	14	49.10	1.09	0.17	2.10	8.94	3794.25	740.87	1.66	1.52	187.28	1.61	4937.07	561.53	20.01
YDR363W	11	31.70	0.55	0.15	1.12	4.61	2036.18	381.91	0.92	0.43	167.83	0.75	2357.36	368.31	10.09
YAR042W	86	29.70	1.05	0.14	1.18	7.54	3472.62	663.58	1.51	0.78	353.69	1.31	5255.89	541.66	12.68
YDR092W	87	45.29	0.93	0.13	1.91	10.94	1851.31	743.24	1.17	1.27	356.24	1.01	4821.09	636.01	13.66

We can see that the first few columns are meta information such as gene ORF and batch id. The rest is the ionomics data.

Data pre-process

The raw data set is needed to be pre-processed. The pre-processing function `PrePro cessing` performs:

- log transformation
- batch correction
- outlier detection
- standardisation

For batch correction, control line could be used. If so, the values belong to control lines are used to be the basis of batch correlation. This data has a control line: **YDL227C** mutant. The code segment below is to identify it:

```
max(with(ion_data, table(Knockout)))
#> [1] 1617
which.max(with(ion_data, table(Knockout)))
```

```
#> YDL227C
#>     209
```

The outlier detection here is univarite method, with a threshold to control the number of outliers. The larger the threshold (`thres_outl`) the more outlier removal.

Standarisation provides a *custom* method. This allows user to use specific std values such as:

```
std <- read.table("../test-data/user_std.tsv", header = T, sep = "\t")
std
#>    Ion     sd
#> 1   Ca 0.1508
#> 2   Cd 0.0573
#> 3   Co 0.0580
#> 4   Cu 0.0735
#> 5   Fe 0.1639
#> 6    K 0.0940
#> 7   Mg 0.0597
#> 8   Mn 0.0771
#> 9   Mo 0.1142
#> 10  Na 0.1075
#> 11  Ni 0.0784
#> 12   P 0.0597
#> 13   S 0.0801
#> 14  Zn 0.0671
```

The pre-process procedure returns not only processed ionomics data but also a symbolic data. This data is based on the inomics data and a threshold(`thres_symb`):

- `0` if ionomics data located between `[-thres_symb, thres_symb]`
- `1` if ionomics data larger than `thres_symb`
- `-1` if ionomics data smaller than `-thres_symb`

The core part of network and enrivhment analysis, clustering, is based on the symbolic data. Note that the symblic data is sensitive to the choices of the threshold.

Let's run the pre-process procedure:

```
pre <- PreProcessing(data = ion_data,
                     var_id = 1, batch_id = 2, data_id = 3,
                     method_norm = "median",
                     control_lines = "YDL227C",
                     control_use = "control",
                     method_outliers = "IQR",
                     thres_outl = 3,
                     stand_method = "std",
```

```
                        stdev = NULL,
                        thres_symb = 3)

names(pre)
#> [1] "stats.raw_data"    "stats.outliers"    "stats.batch_data"
#> [4] "data.long"         "data.gene.logFC"   "data.gene.zscores"
#> [7] "data.gene.symb"    "plot.dot"          "plot.hist"
```

The results includes summaries of raw data and processed data. The latter is:

```
pre$stats.batch_data %>%
  kable(caption = 'Processed data summary', digits = 2, booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10)
```

Table 2: Processed data summary

Ion	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Variance
Ca	-4.45	-0.28	-0.13	-0.12	0.02	2.35	0.11
Cd	-1.70	0.03	0.10	0.11	0.17	0.93	0.03
Co	-2.80	0.02	0.09	0.06	0.15	1.60	0.05
Cu	-0.66	-0.10	-0.03	-0.01	0.04	5.28	0.04
Fe	-7.48	-0.17	-0.06	-0.02	0.07	6.88	0.14
K	-2.21	-0.17	-0.01	-0.08	0.09	1.83	0.08
Mg	-1.84	-0.06	0.01	-0.01	0.07	1.69	0.03
Mn	-4.11	-0.24	-0.08	-0.13	0.01	1.78	0.06
Mo	-2.03	-0.26	-0.08	-0.08	0.09	4.44	0.13
Na	-7.41	-0.53	-0.22	-0.33	-0.04	1.25	0.24
Ni	-2.40	-0.01	0.09	0.12	0.21	7.90	0.12
P	-1.18	-0.06	0.00	-0.01	0.06	1.45	0.02
S	-2.38	-0.03	0.05	0.06	0.16	2.38	0.04
Zn	-0.46	-0.08	-0.03	-0.01	0.03	4.60	0.02

The pre-processed data and its symbolic data are like like:

```
pre$data.gene.zscores %>% head() %>%
  kable(caption = 'Pre-processed data', digits = 2, booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

```
pre$data.gene.symb %>% head() %>%
  kable(caption = 'Symbolic data', booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10)
```

The symbolic data is calulated from processed data with control of `thres_symb` (here is 3). You can obtain a new symbol data by assigning new threshold to the function `symbol_data`:

Table 3: Pre-processed data

Line	Ca	Cd	Co	Cu	Fe	K	Mg	Mn	Mo	Na	Ni	P	S	Zn
YAL004W	-1.16	0.75	1.19	-0.47	0.04	0.61	0.51	-0.84	-0.08	-1.84	1.71	0.52	0.33	-0.09
YAL005C	-1.67	0.84	0.55	0.58	-2.79	0.59	0.31	-1.16	-1.42	-0.12	1.48	0.73	0.13	-0.13
YAL007C	-2.12	0.64	0.23	-0.53	-0.24	0.79	-0.09	-0.14	1.22	-0.92	0.00	0.09	-0.29	-0.65
YAL008W	-2.34	1.13	0.21	-0.73	-2.16	0.52	-0.02	-0.87	0.93	-0.58	0.02	-0.09	-0.73	-0.47
YAL009W	-1.18	0.66	0.55	-1.11	-3.91	0.22	0.09	-0.18	1.50	-0.84	-0.09	0.14	0.01	-0.36
YAL010C	-1.28	1.43	2.27	0.46	1.53	-2.75	0.04	-0.74	-9.71	-4.30	2.42	-0.98	-0.05	-0.01

Table 4: Symbolic data

Line	Ca	Cd	Co	Cu	Fe	K	Mg	Mn	Mo	Na	Ni	P	S	Zn
YAL004W	0	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL005C	0	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL007C	0	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL008W	0	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL009W	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
YAL010C	0	0	0	0	0	0	0	0	-1	-1	0	0	0	0

```
data_symb <- symbol_data(pre$data.gene.zscores, thres_symb = 2)
data_symb %>% head() %>%
  kable(caption = 'Symbolic data with threshold of 2', booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10)
```

Table 5: Symbolic data with threshold of 2

Line	Ca	Cd	Co	Cu	Fe	K	Mg	Mn	Mo	Na	Ni	P	S	Zn
YAL004W	0	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL005C	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
YAL007C	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
YAL008W	-1	0	0	0	-1	0	0	0	0	0	0	0	0	0
YAL009W	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
YAL010C	0	0	1	0	0	-1	0	0	-1	-1	1	0	0	0

The pre-processed data distribution is:

```
pre$plot.hist
```

Data filtering

There are a lot of ways to filter gene. Here we filter gene based on symbolic data:

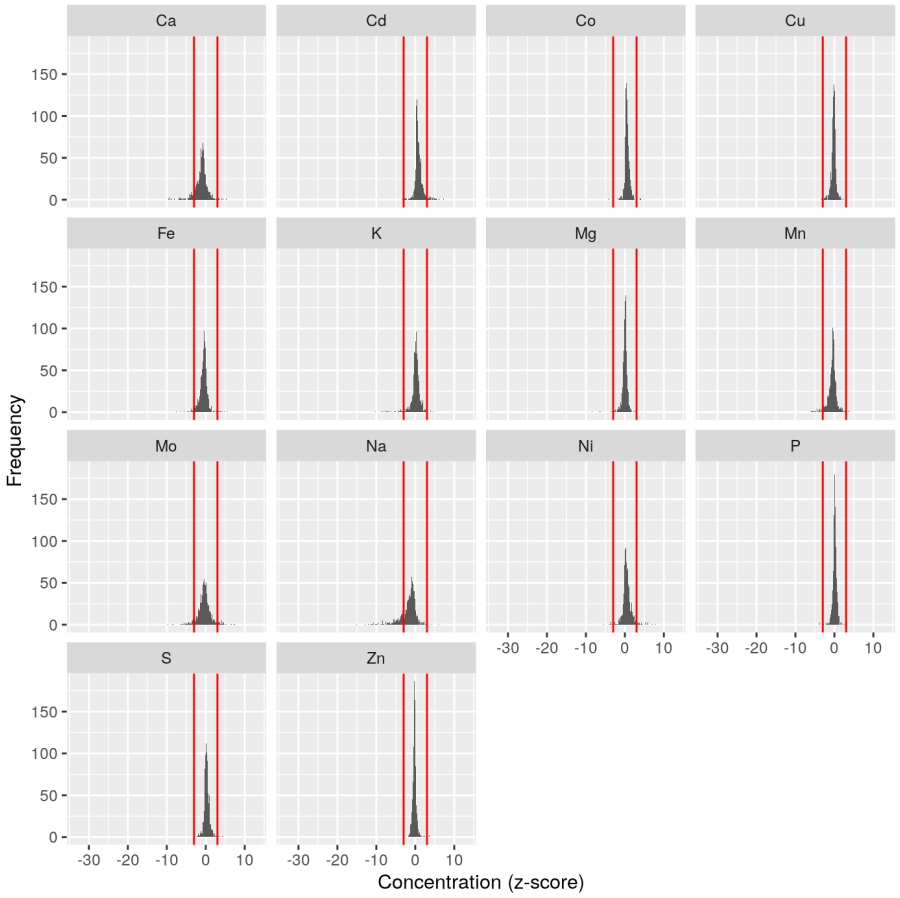

Figure 1: Ionome data distribution plot

```
data <- pre$data.gene.zscores
data_symb <- pre$data.gene.symb
idx <- rowSums(abs(data_symb[, -1])) > 0
dat <- data[idx, ]
dat_symb <- data_symb[idx, ]
dim(dat)
#> [1] 549  15
```

Data clustering

The hierarchical cluster analysis is the key part of gene network and gene enrichment analysis. The methodology is as follow:

- Compute the distance of symbolic data
- Hierarchical cluster analysis on the distance
- Identify clusters/groups with a threshold of minimal number of cluster size

One example is:

```
clust <- gene_clus(dat_symb[, -1], min_clust_size = 10)
names(clust)
#> [1] "clus"    "idx"     "tab"     "tab_sub"
```

The cluster centres are:

```
clust$tab_sub
#>   cluster nGenes
#> 1       4    149
#> 2      11     72
#> 3       7     36
#> 4       1     27
#> 5      18     15
#> 6       5     12
#> 7       3     11
#> 8       8     11
```

It indicates that clusters and their number of genes (larger than `min_cluster_size`).

Gene network

The gene network uses both the ionomics and symboloc data. The similarity measure on the ionomics data is filtered by the similarity threshold located between 0 and 1, and cluster centres of symbolic data. The filter values are then used for network analysis.

The similarity measure method is one of *pearson*, *spearman*, *kendall*, *cosine*, *mahal_cosine* or *hybrid_mahal_cosine*.

First, the Pearson correlation is used to build up the network:

```
net <- GeneNetwork(data = dat,
                   data_symb = dat_symb,
                   min_clust_size = 10,
                   thres_corr = 0.75,
                   method_corr = "pearson")
```

The network with nodes colouring by the symbolic clustering is:

```
net$plot.pnet1
```

The same network, but nodes are colured by the netwok community detection:

```
net$plot.pnet2
```

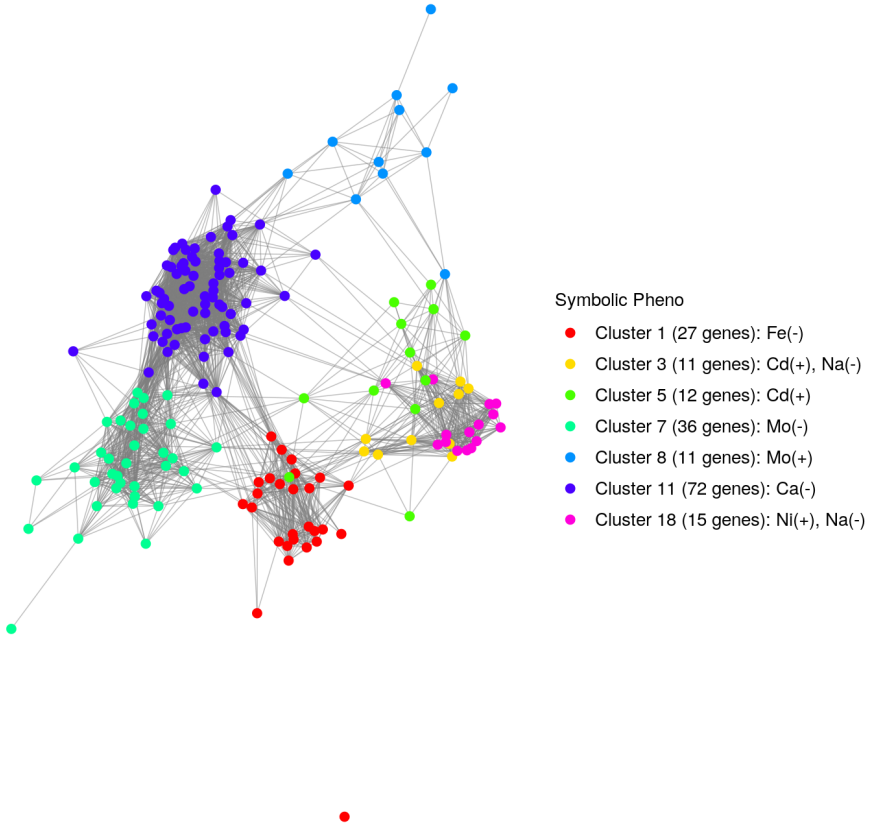

Figure 2: **Netwok analysis based on Pearson correlation: symbolic clustering**

The network analysis also returns a network impact and betweeness plot:

```
net$plot.impact_betweenness
```

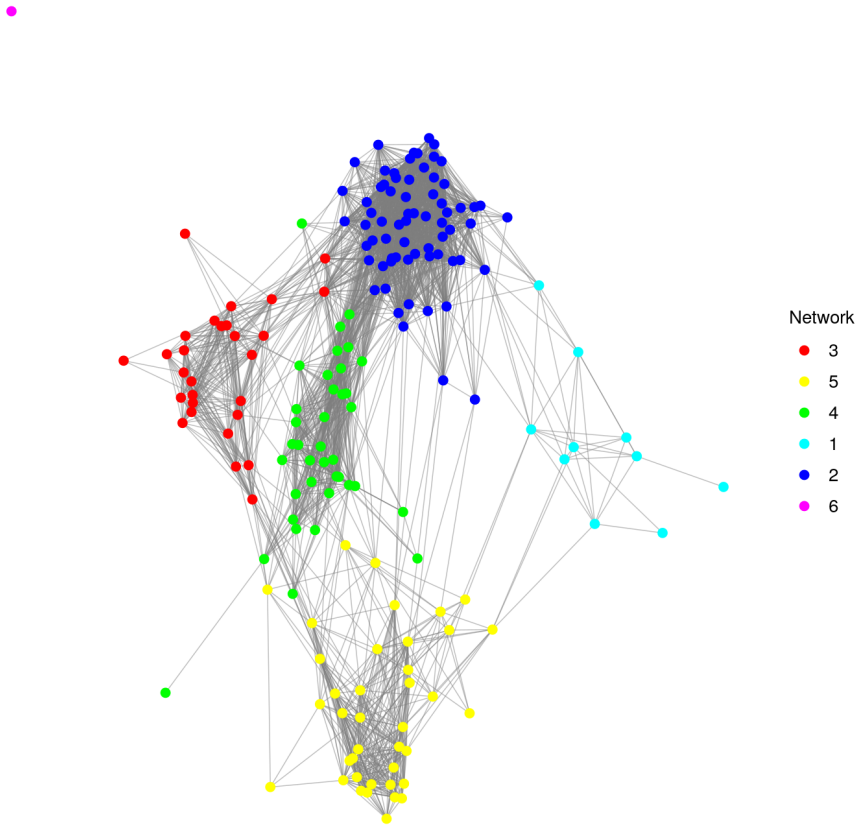

Figure 3: Netwok analysis based on Pearson correlation: community detction

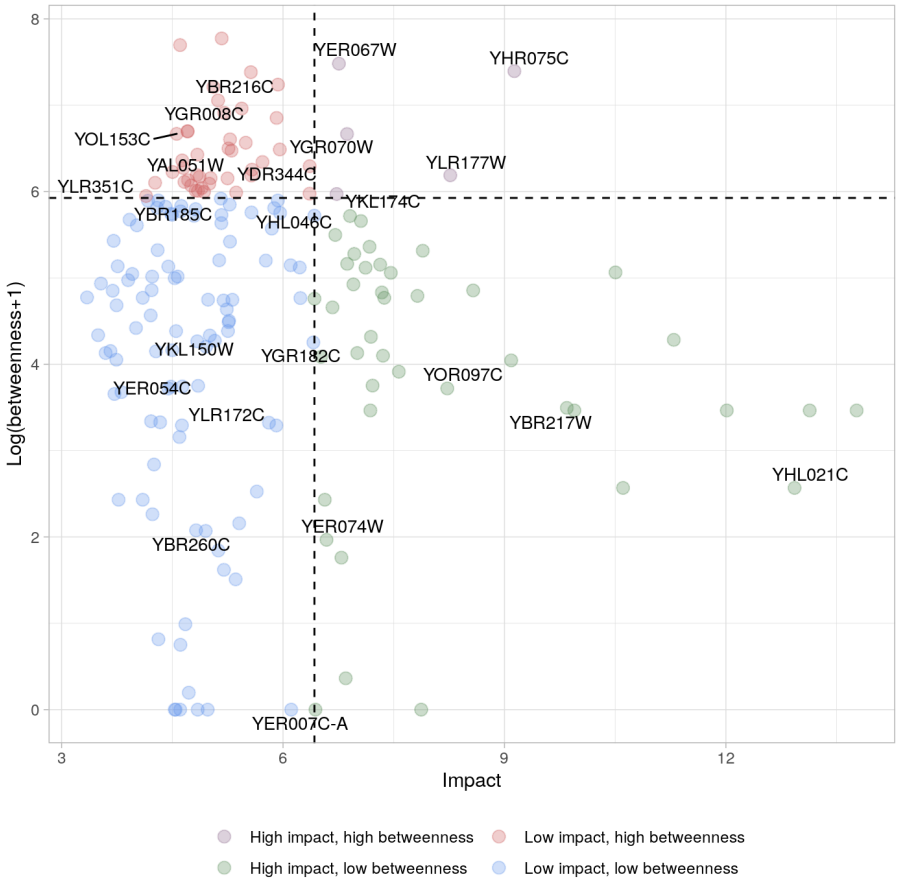

Figure 4: Netwok analysis based on Pearson correlation: impact and betweeness

For the comparision purpose, we use different similarity methods. Here use *Cosine*:

```
net_1 <- GeneNetwork(data = dat,
                     data_symb = dat_symb,
                     min_clust_size = 10,
                     thres_corr = 0.75,
                     method_corr = "cosine")
net_1$plot.pnet1
```

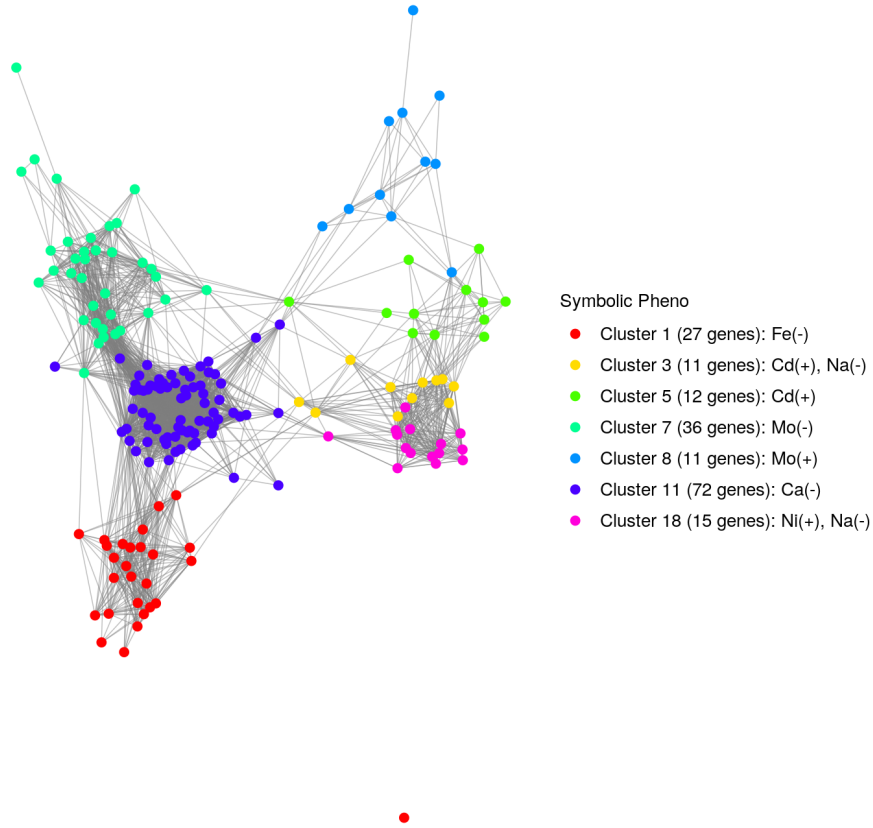

Figure 5: Netwok analysis based on Cosine

```
net_1$plot.pnet2
```

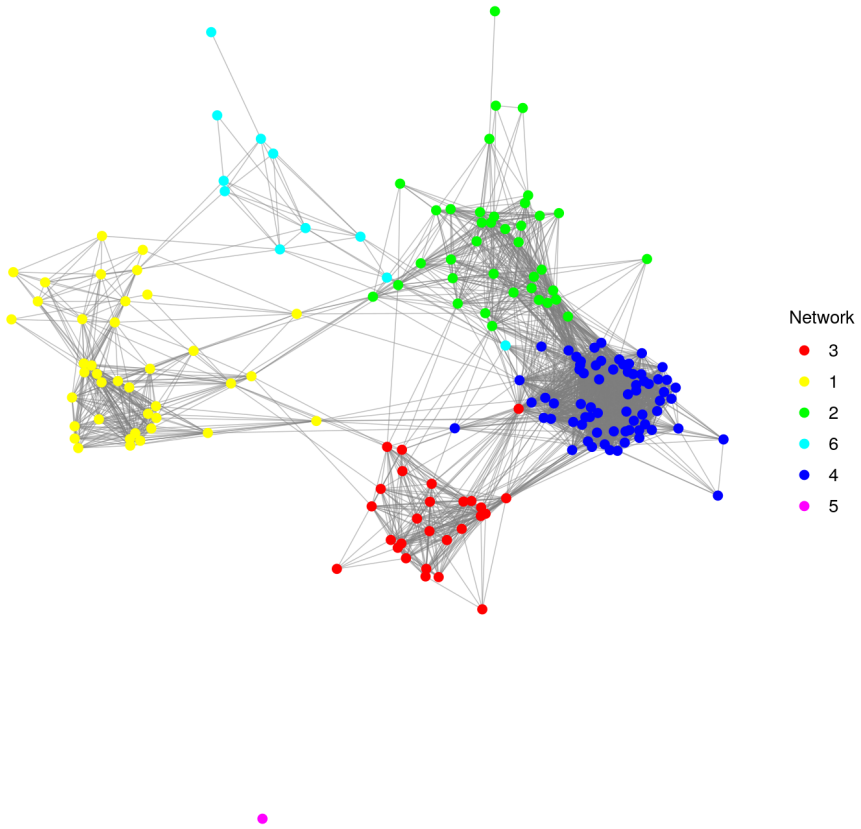

Figure 6: Netwok analysis based on Cosine

Use *Hybrid Mahalanobis Cosine*:

```
net_2 <- GeneNetwork(data = dat,
                     data_symb = dat_symb,
                     min_clust_size = 10,
                     thres_corr = 0.75,
                     method_corr = "mahal_cosine")
net_2$plot.pnet1
```

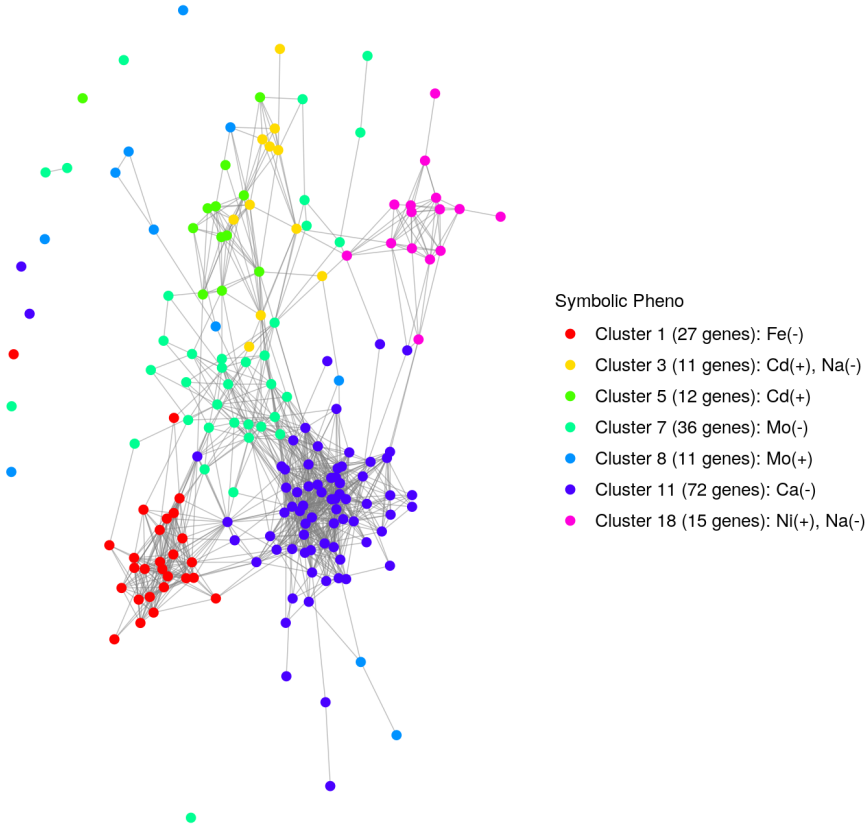

Figure 7: Netwok analysis based on Mahalanobis Cosine

```
net_2$plot.pnet2
```

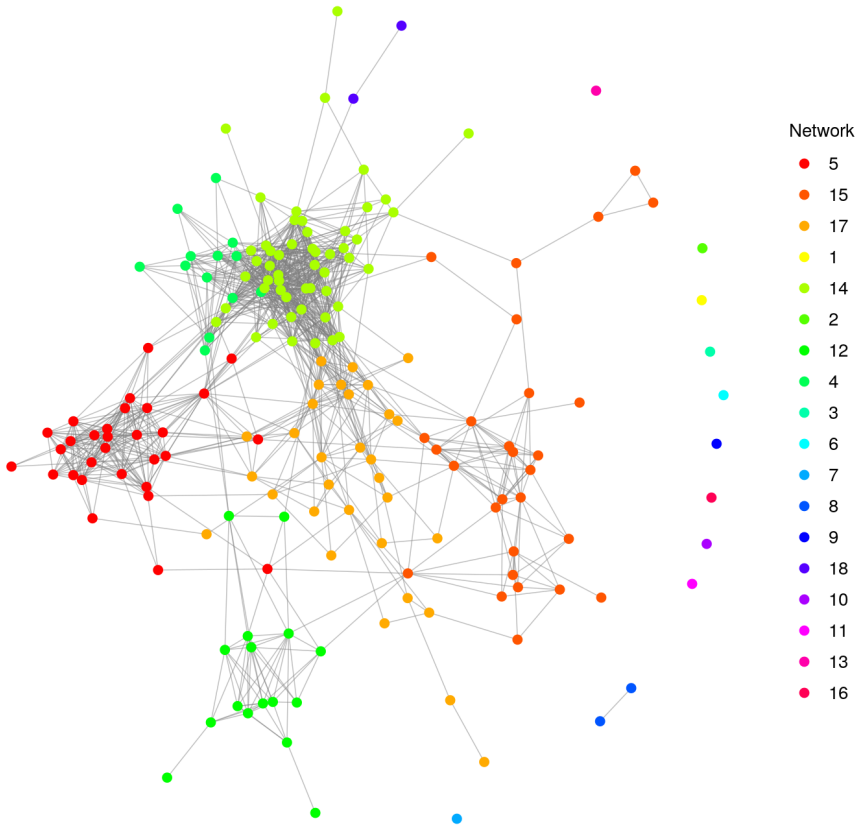

Figure 8: Netwok analysis based on Mahalanobis Cosine

Again, we use *Hybrid Mahalanobis Cosine*:

```
net_3 <- GeneNetwork(data = dat,
                     data_symb = dat_symb,
                     min_clust_size = 10,
                     thres_corr = 0.75,
                     method_corr = "hybrid_mahal_cosine")
net_3$plot.pnet1
```

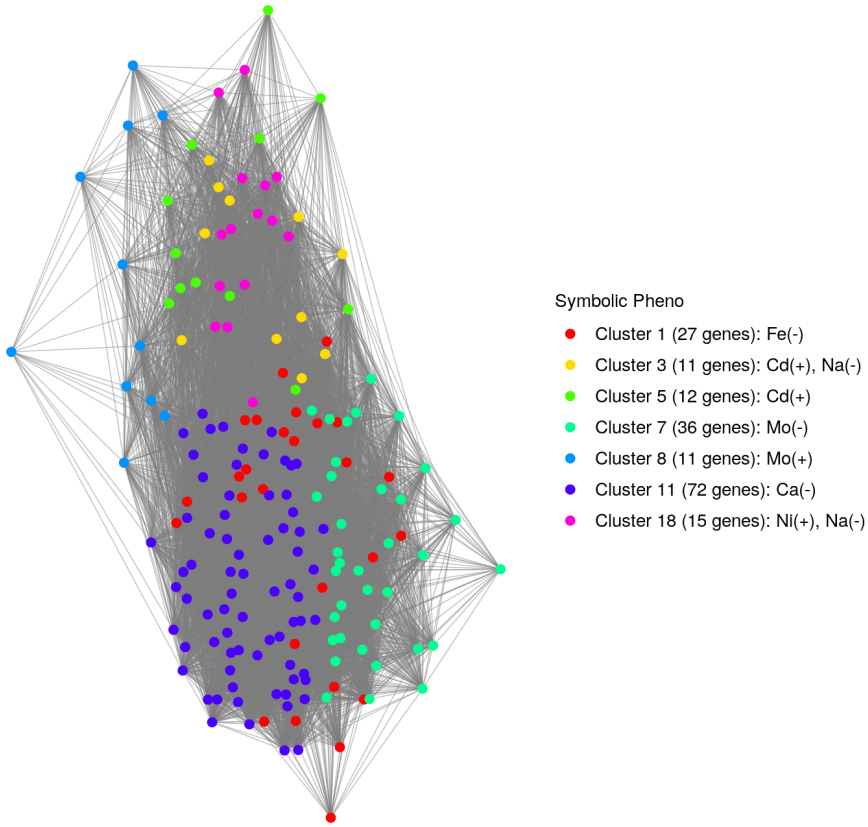

Figure 9: Netwok analysis based on Hybrid Mahalanobis Cosine

```
net_3$plot.pnet2
```

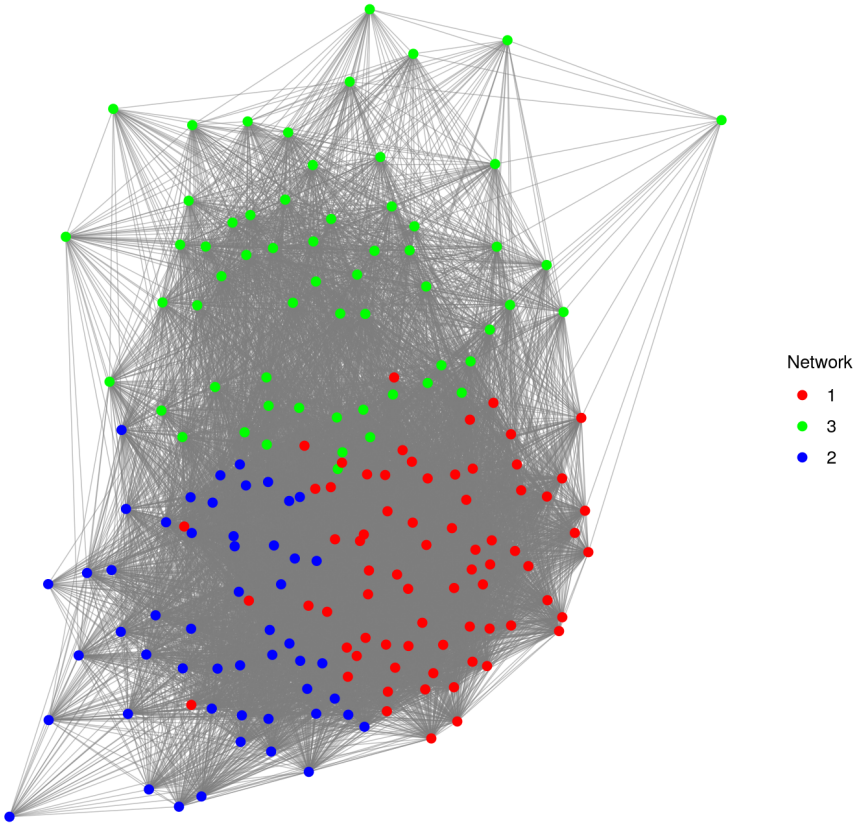

Figure 10: Netwok analysis based on Hybrid Mahalanobis Cosine

Enrichment analysis

The KEGG enrichment analysis:

```
kegg <- kegg_enrich(data = dat_symb, min_clust_size = 10, pval = 0.05,
                    annot_pkg =  "org.Sc.sgd.db")

#' kegg
kegg %>%
  kable(caption = 'KEGG enrichmenat analysis', digits = 3, booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

Table 6: KEGG enrichmenat analysis

Cluster	KEGGID	Pvalue	Count	Size	Term
Cluster 7 (36 genes)	03010	0.029	9	16	Ribosome
Cluster 7 (36 genes)	00330	0.031	3	3	Arginine and proline metabolism
Cluster 18 (15 genes)	00290	0.009	2	2	Valine, leucine and isoleucine biosynthesis
Cluster 18 (15 genes)	00520	0.009	2	2	Amino sugar and nucleotide sugar metabolism
Cluster 18 (15 genes)	00260	0.012	3	6	Glycine, serine and threonine metabolism
Cluster 18 (15 genes)	00010	0.024	2	3	Glycolysis / Gluconeogenesis
Cluster 18 (15 genes)	01110	0.037	5	22	Biosynthesis of secondary metabolites
Cluster 3 (11 genes)	00400	0.009	2	2	Phenylalanine, tyrosine and tryptophan biosynthesis
Cluster 8 (11 genes)	01100	0.006	6	55	Metabolic pathways
Cluster 8 (11 genes)	00564	0.027	2	6	Glycerophospholipid metabolism

Note that there can be none results for KRGG enrichment analysis. Change arguments such as `thres_clus` as appropriate.

The GO Terms enrichment analysis:

```
go <- go_enrich(data = dat_symb, min_clust_size = 10, pval = 0.05,
                ont = "BP", annot_pkg =  "org.Sc.sgd.db")
#' go
go %>% head() %>%
  kable(caption = 'GO Terms enrichmenat analysis', digits = 3, booktabs = T) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

Table 7: GO Terms enrichmenat analysis

Cluster	ID	Description	Pvalue	Count	CountUniverse	Ontology
Cluster 4 (149 genes)	GO:0051336	regulation of hydrolase activity	0.0018	4	12	BP
Cluster 4 (149 genes)	GO:0043085	positive regulation of catalytic activity	0.0044	4	15	BP
Cluster 4 (149 genes)	GO:0035303	regulation of dephosphorylation	0.0068	2	3	BP
Cluster 4 (149 genes)	GO:0046889	positive regulation of lipid biosynthetic process	0.0068	2	3	BP
Cluster 4 (149 genes)	GO:1903727	positive regulation of phospholipid metabolic process	0.0068	2	3	BP
Cluster 4 (149 genes)	GO:0044764	multi-organism cellular process	0.0074	3	9	BP

Exploratory analysis

Some analysis are performed in terms of ions, i.e. feature, including PCA and correlation.

```
expl <- ExploratoryAnalysis(data = dat)
```

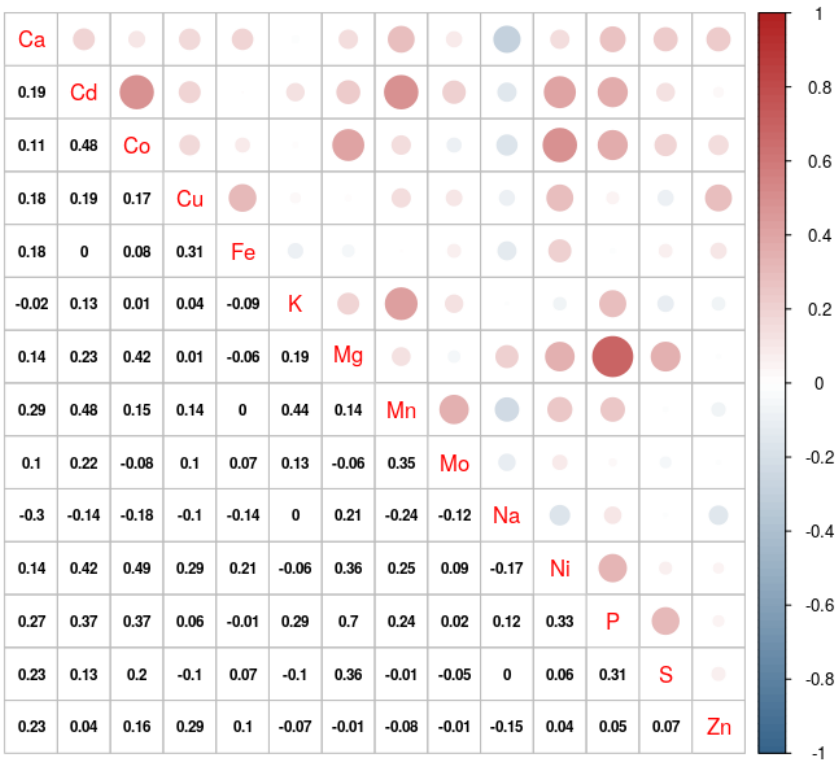

Figure 11: Exploratory analysis plots with respect to ionome

```
expl$plot.PCA_Individual
```

```
expl$plot.correlation_network
```

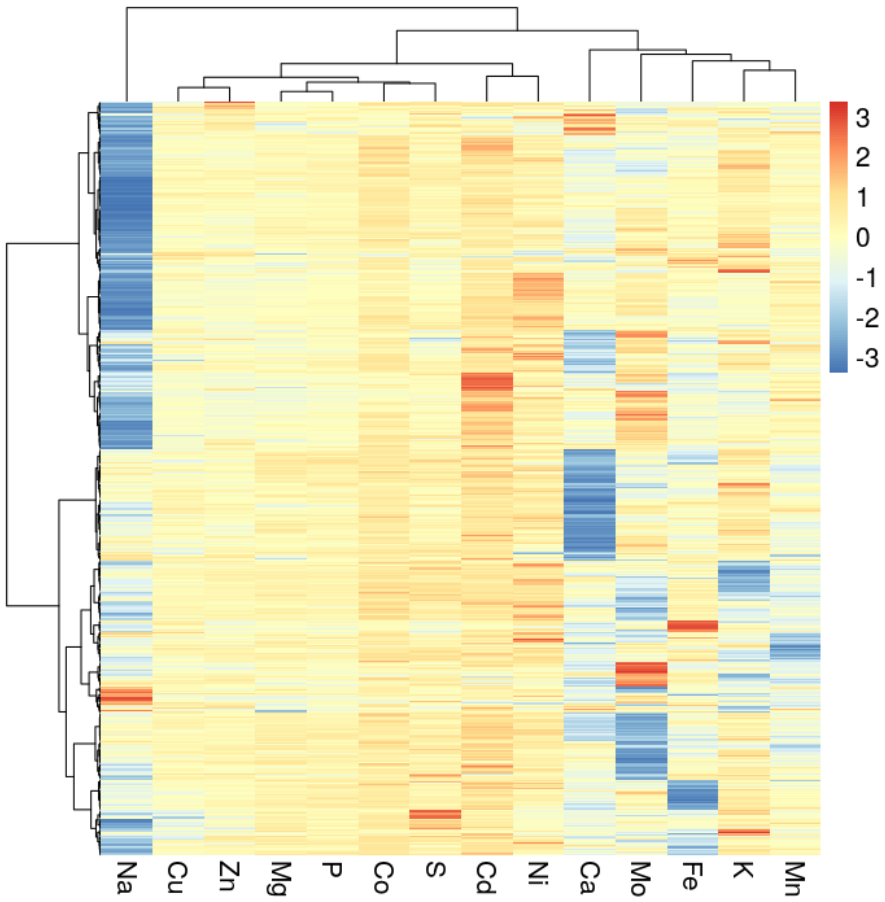

Figure 12: Exploratory analysis plots with respect to ionome

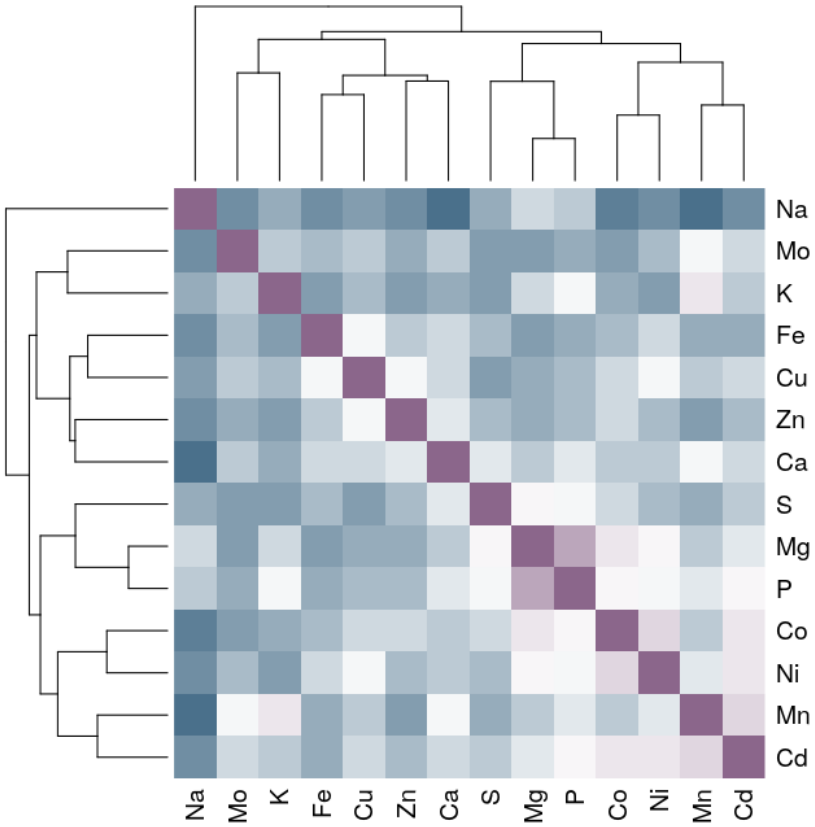

Figure 13: Exploratory analysis plots with respect to ionome

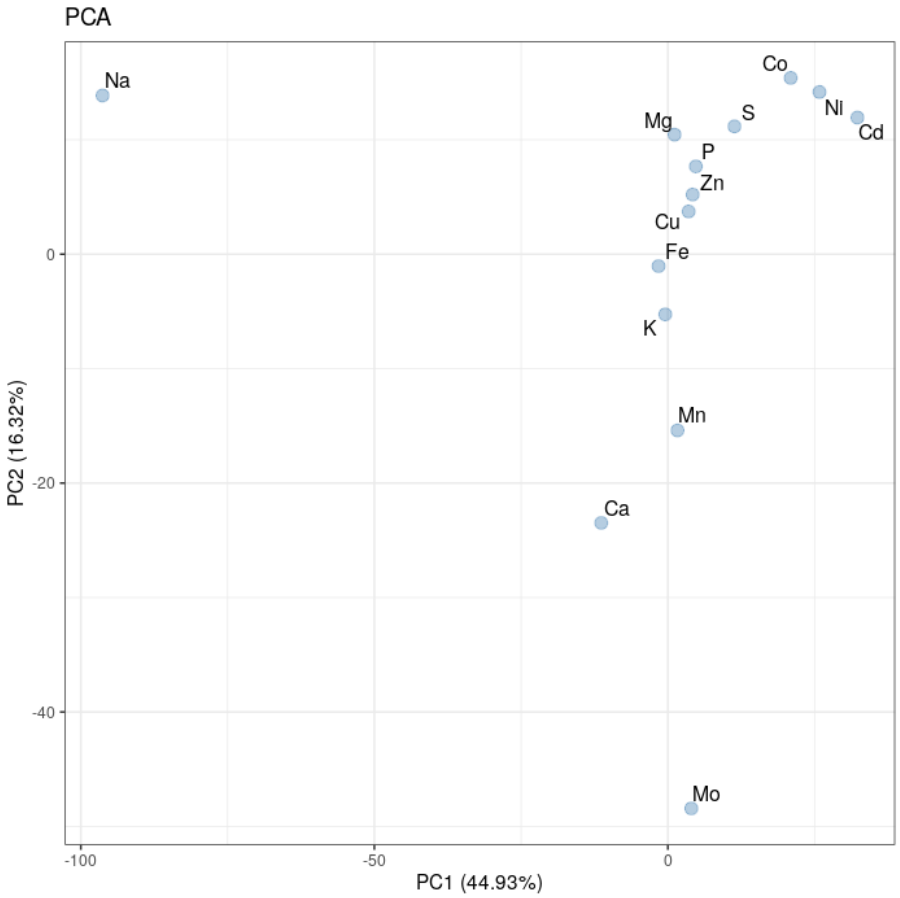

Figure 14: Exploratory analysis plots with respect to ionome

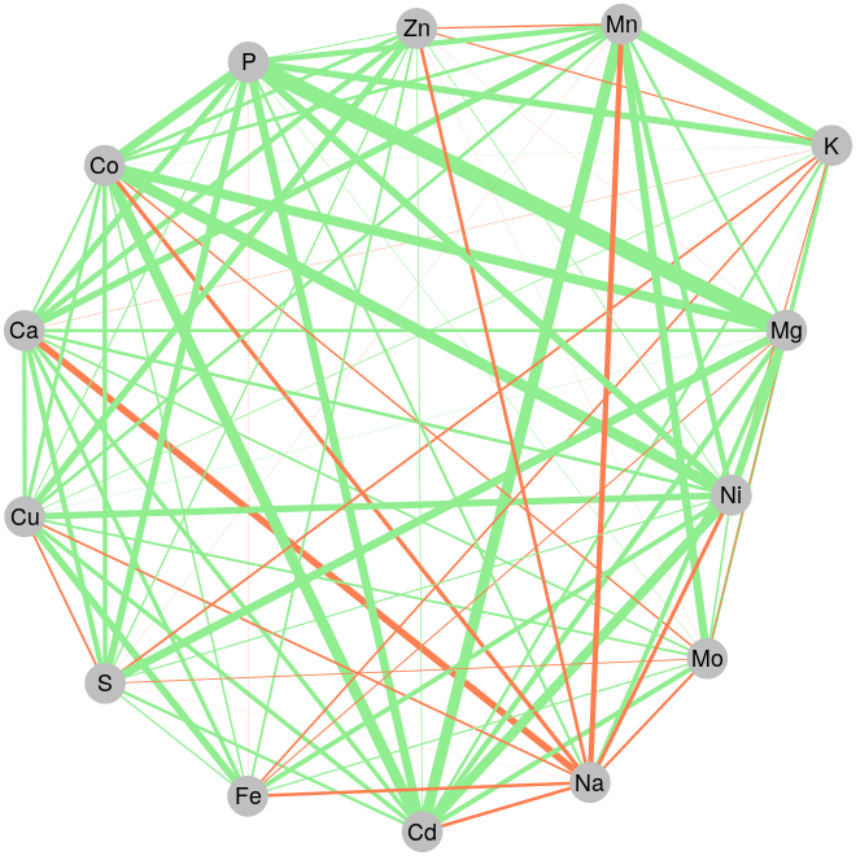

Figure 15: Exploratory analysis plots with respect to ionome