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
YLR396C	63	36.25	1.01	0.23	1.47	7.72	1931.25	963.00	0.84	0.71	65.87	1.99	4868.28	671.17	17.33
YBR183W	4	26.28	0.85	0.13	1.15	3.01	1643.04	176.80	0.81	0.49	47.51	0.59	1384.92	202.11	11.54
YER032W	13	63.37	0.91	0.15	1.40	4.37	3276.39	682.44	1.35	1.47	166.13	1.29	4425.86	544.39	17.54
YLR273C	30	35.80	1.16	0.15	1.51	5.19	1962.31	669.62	1.24	0.75	226.15	1.40	4430.67	537.62	13.99
YJL163C	64	35.72	0.90	0.14	1.30	6.28	2743.31	782.05	1.20	1.61	324.65	0.94	4912.65	523.57	16.09
YDL227C	4	36.20	0.79	0.13	1.35	5.99	1838.02	280.08	0.97	0.60	121.33	0.67	2161.06	354.26	14.09
YDL227C	2	25.55	0.79	0.15	1.88	12.22	2335.82	273.74	1.08	1.07	125.34	1.13	2288.50	167.67	14.95
YGL222C	32	34.83	0.94	0.20	1.39	7.30	2551.78	683.13	1.17	1.02	213.52	1.28	4967.27	524.68	16.73
YGL205W	32	46.46	1.26	0.21	1.32	8.48	2604.82	580.45	1.35	1.78	169.46	1.08	4372.83	466.35	15.15
YDL227C	5	20.23	0.84	0.16	1.47	3.77	2768.37	479.71	1.33	0.77	195.37	0.69	3308.30	347.62	15.73

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. This involves:

- 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
```

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 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.

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`

Ths symbolic data is important since the network analysis will use it along with ionomics data, and enrichment analysis will be performd only based on it. It also should be noted 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 pre-processed data are returned with some summaries, one of them is the z-score.

```
pre$stats.batch_data %>% kable(caption = 'Summary: z-scores', digits = 2) %>%
  kable_styling(full_width = F, font_size = 10)
```

Table 2: Summary: z-scores

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) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

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

```
pre$data.gene.symb %>% head() %>%
  kable(caption = 'Symbolic data') %>%
  kable_styling(full_width = F, font_size = 10)
```

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

The pre-processed data distribution is:

```
pre$plot.hist
```

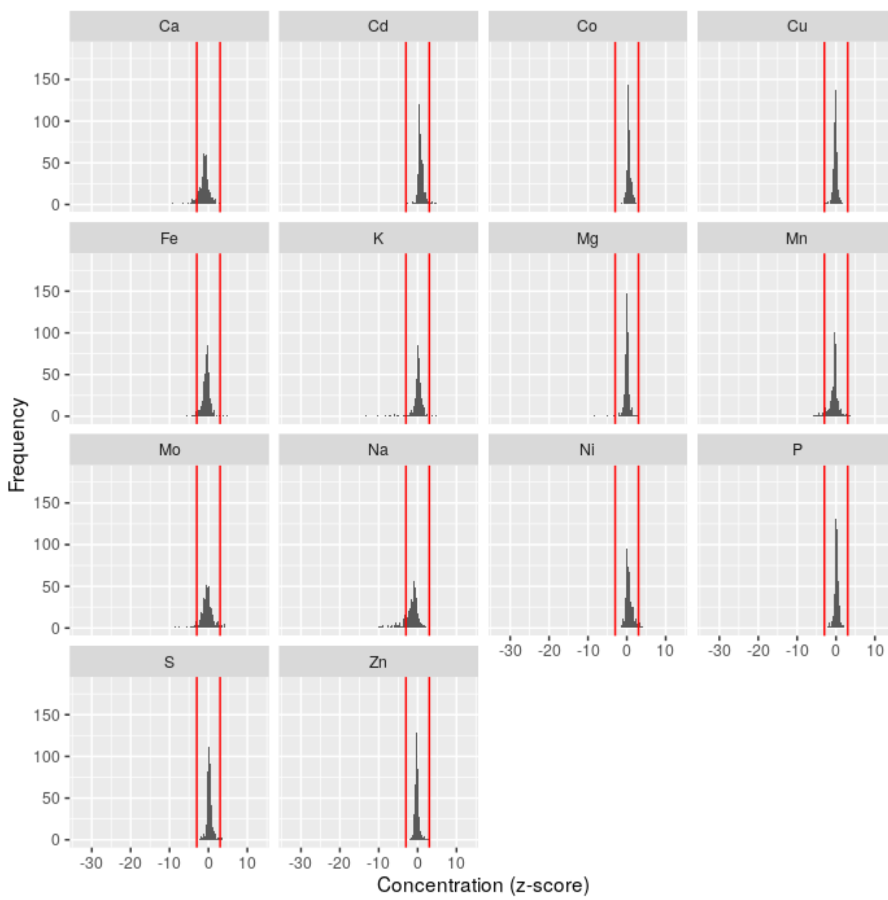

Figure 1: Ionome data distribution plot

Data filtering

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

```
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
```

Gene network

The gene network is based on the ionomics and symboloc data and uses the similarity measure results to build up the network. 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")
net$plot.pnet1
```

```
net$plot.pnet2
```

```
net$plot.impact_betweenness
```

The node colours are indicated by either the similarity measures or the network community detection, i.e. clustering.

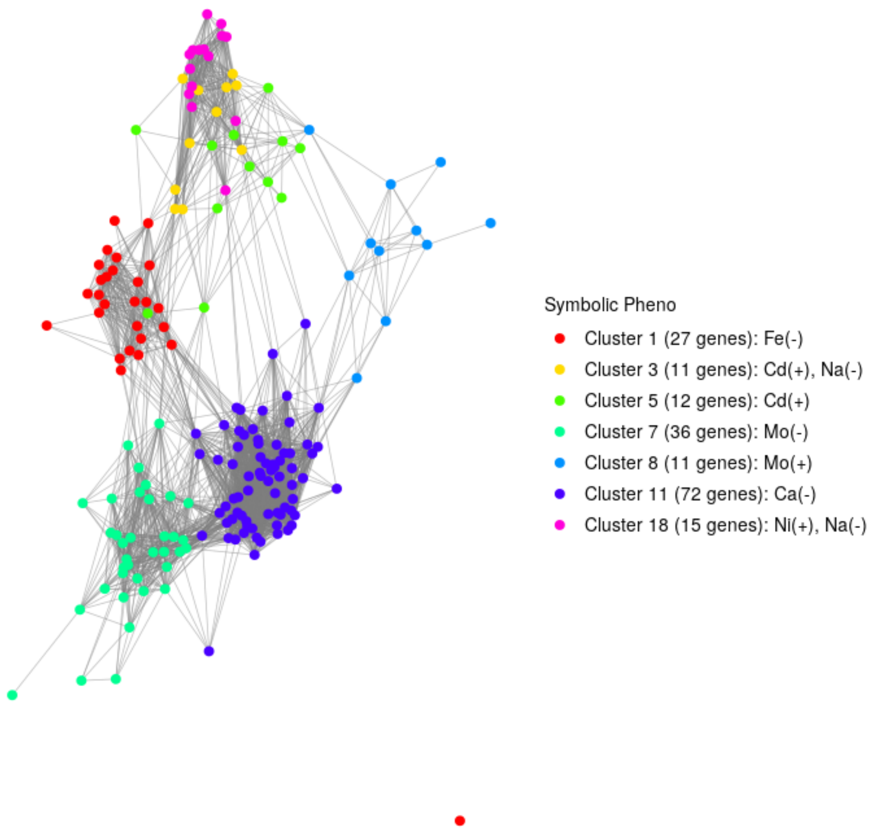

Figure 2: **Netwok analysis based on Pearson correlation**

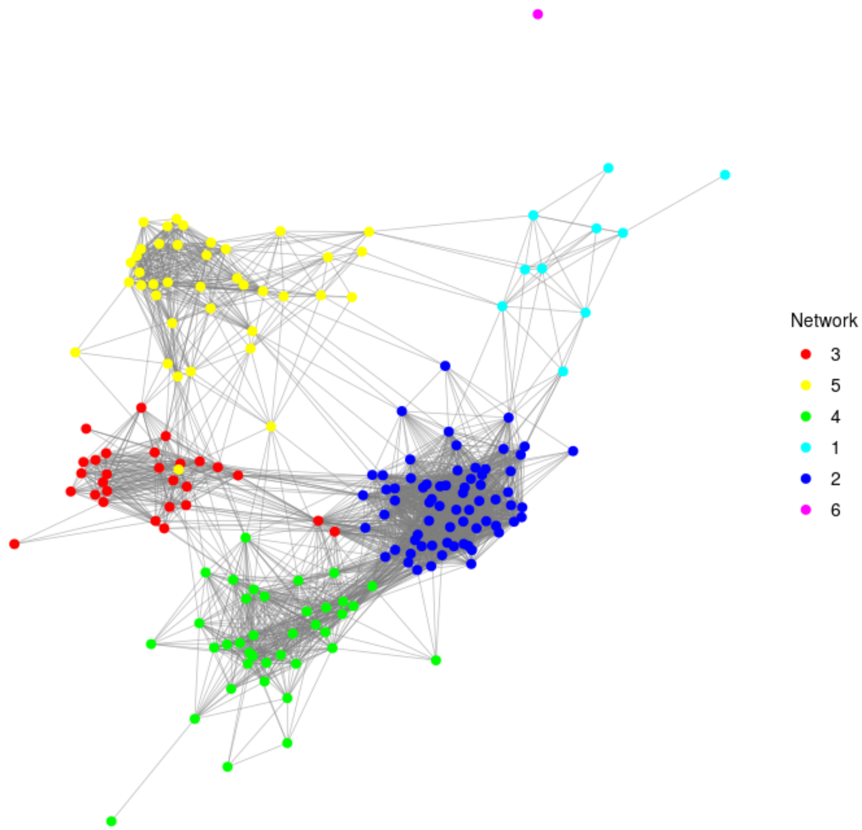

Figure 3: Netwok analysis based on Pearson correlation

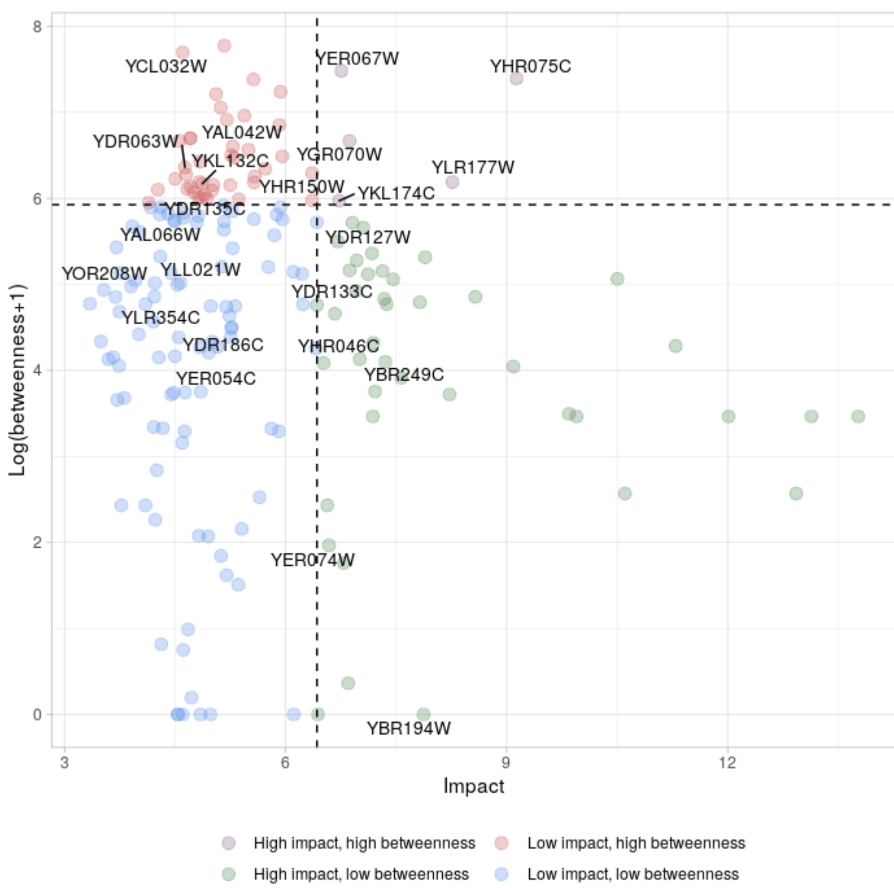

Figure 4: Netwok analysis based on Pearson correlation

For the comparision, 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
```

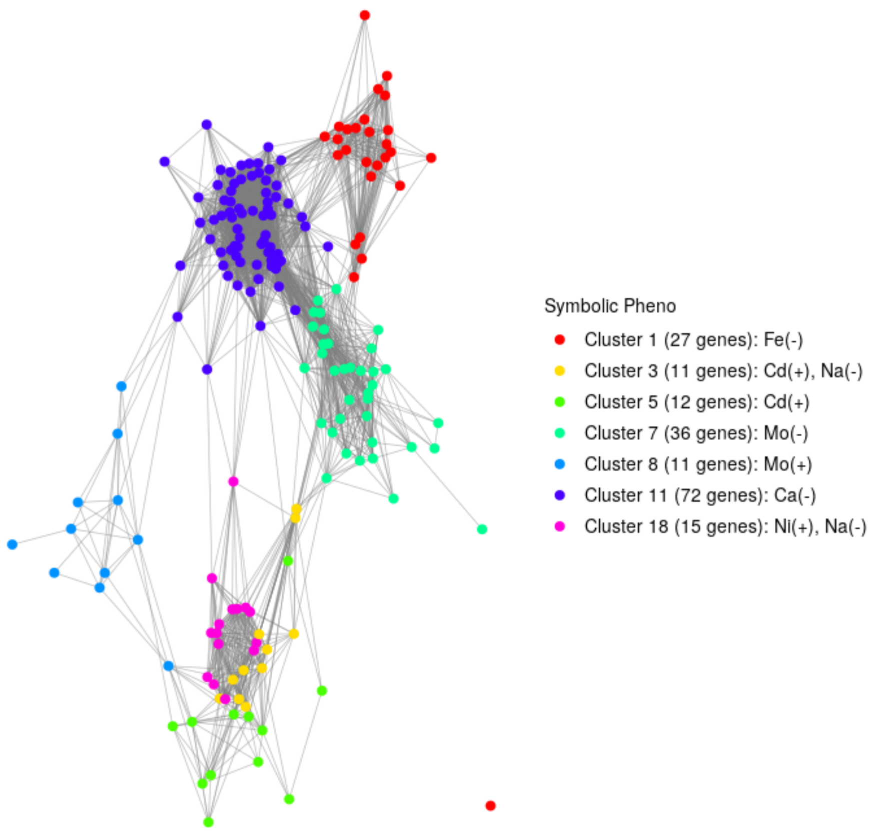

Figure 5: **Netwok analysis based on Cosine**

```
net_1$plot.pnet2
```

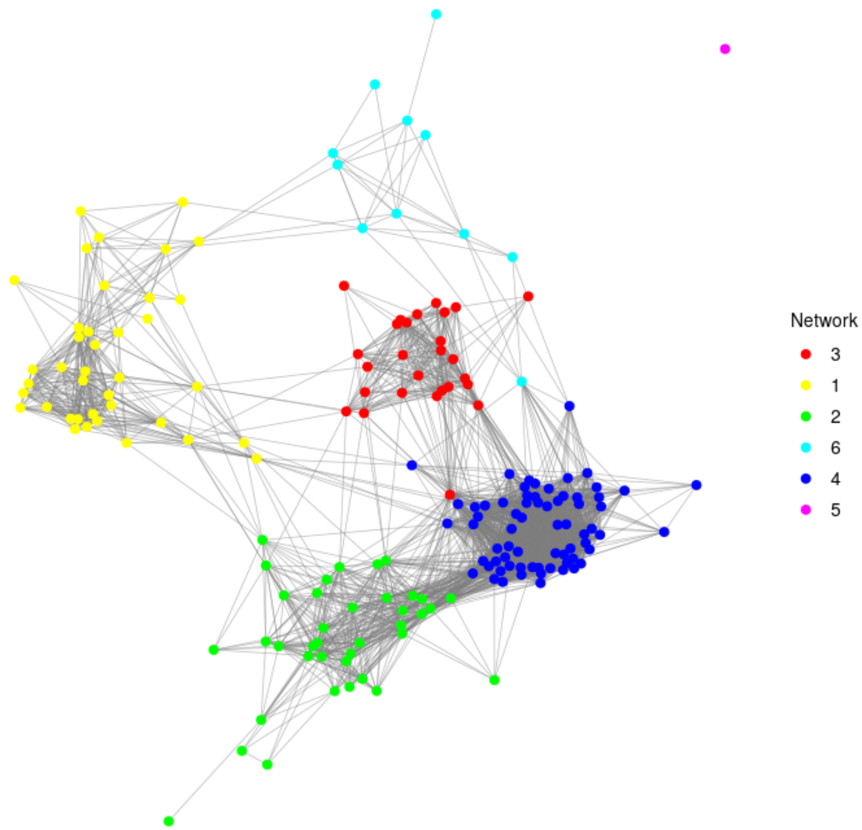

Figure 6: Netwok analysis based on Cosine

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
```

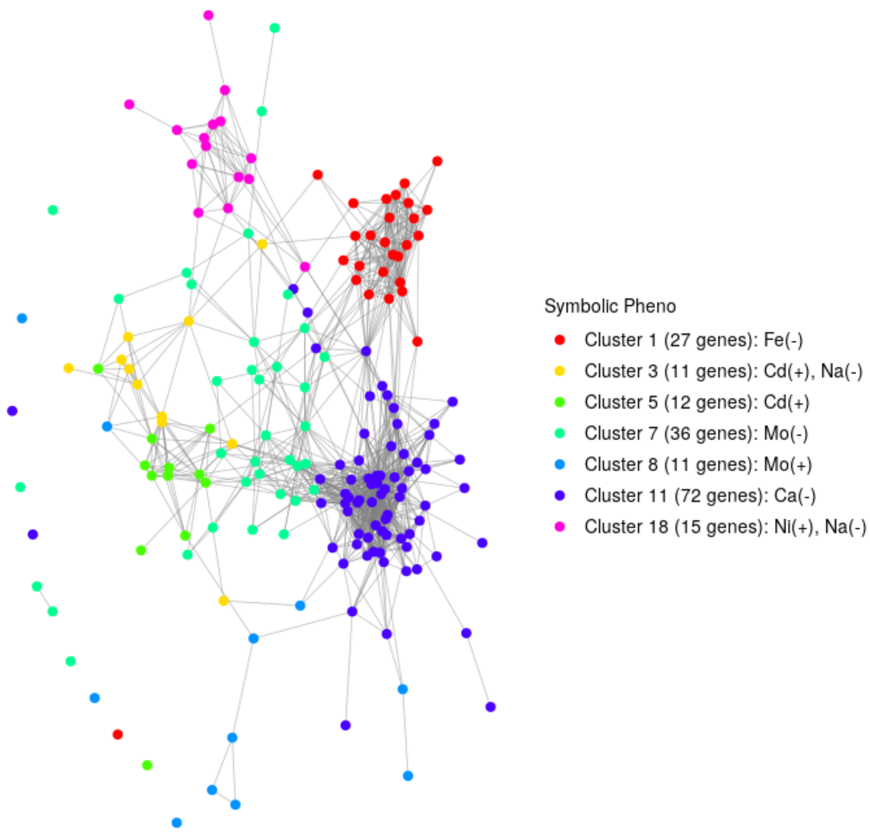

Figure 7: **Netwok analysis based on Mahalanobis Cosine**

```
net_2$plot.pnet2
```

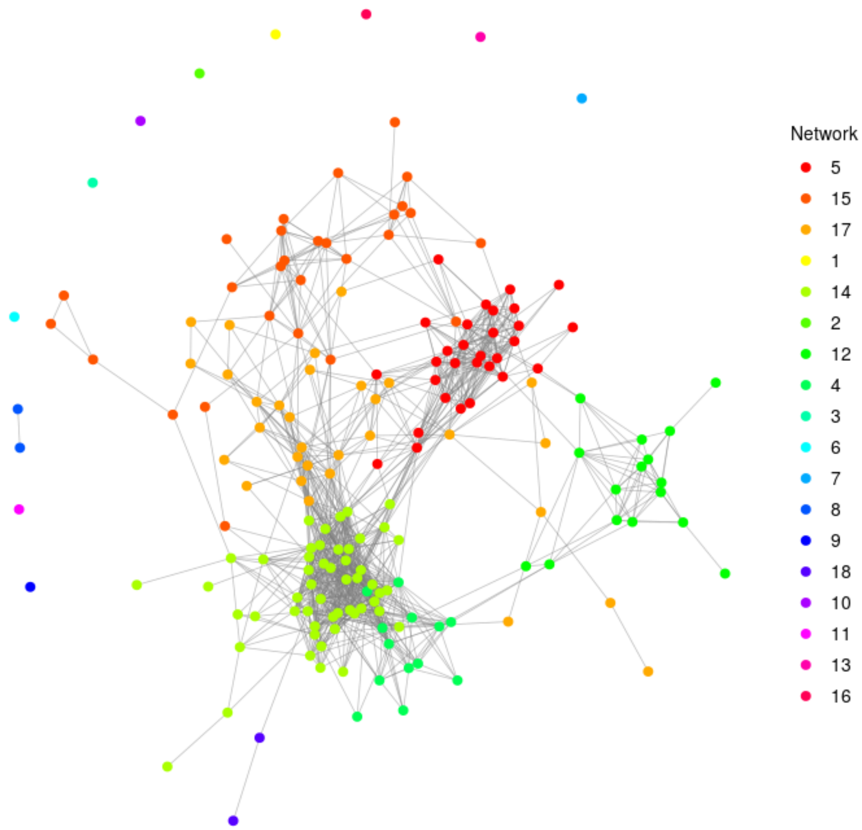

Figure 8: Netwok analysis based on Mahalanobis Cosine

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
```

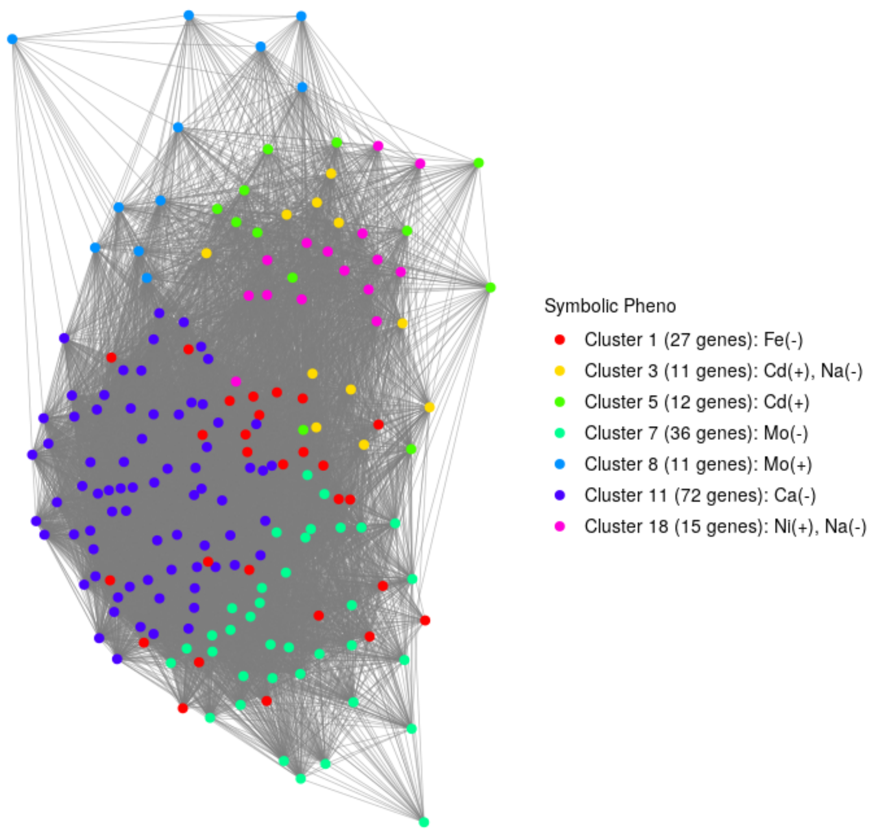

Figure 9: **Netwok analysis based on Hybrid Mahalanobis Cosine**

```
net_3$plot.pnet2
```

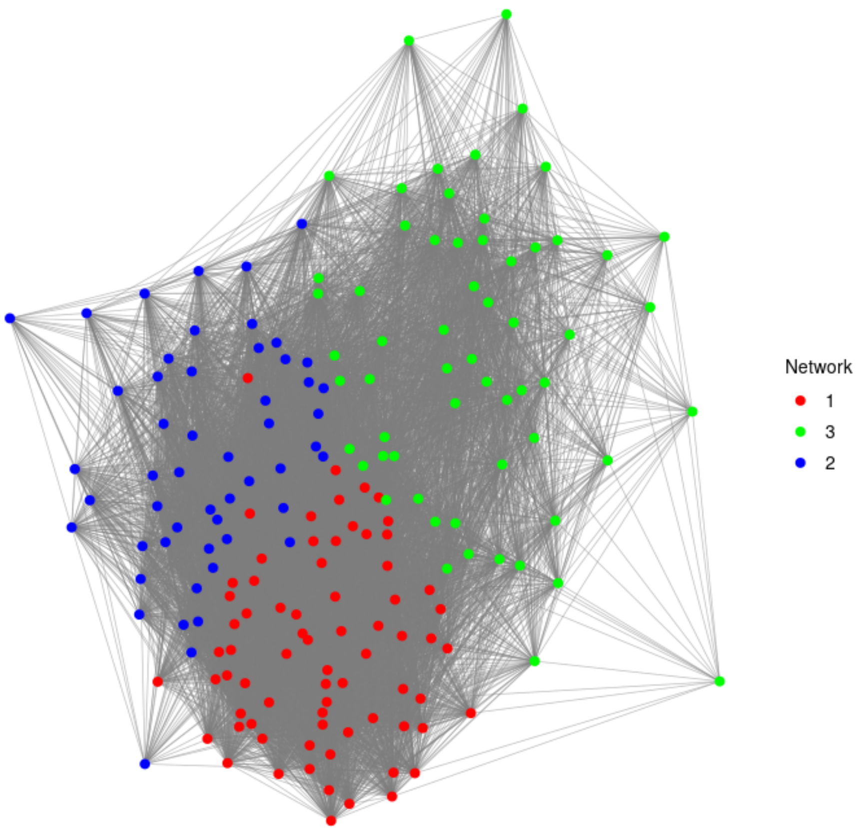

Figure 10: Netwok analysis based on Hybrid Mahalanobis Cosine

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) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

Table 5: 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) %>%
  kable_styling(full_width = F, font_size = 10,
                latex_options = c("striped", "scale_down"))
```

Table 6: 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)
```

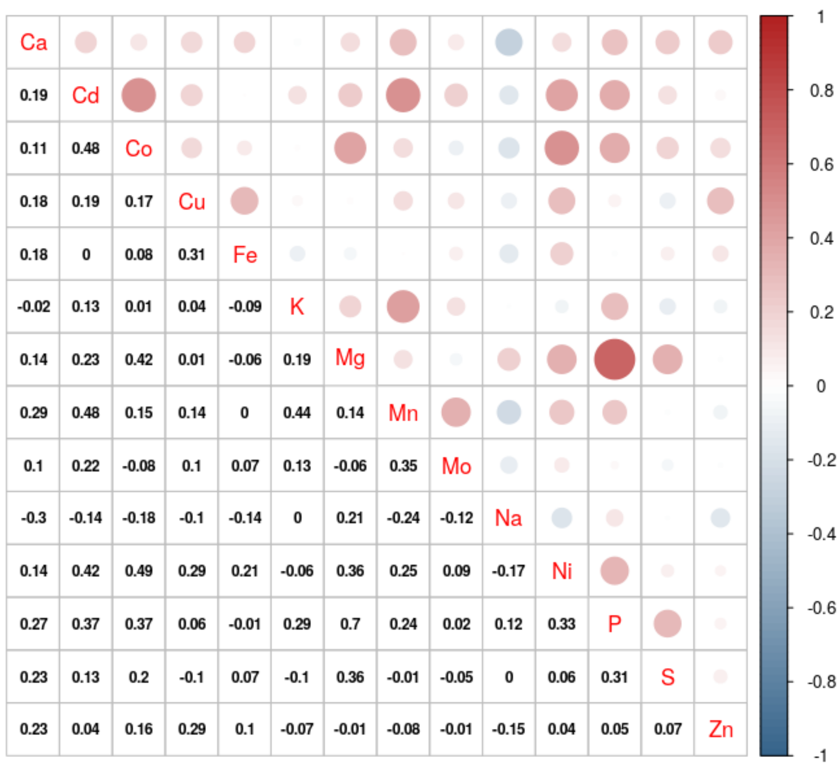

Figure 11: Exploratory analysis plots with respect to ionome

```
expl$plot.PCA_Individual
```

```
expl$plot.correlation_network
```

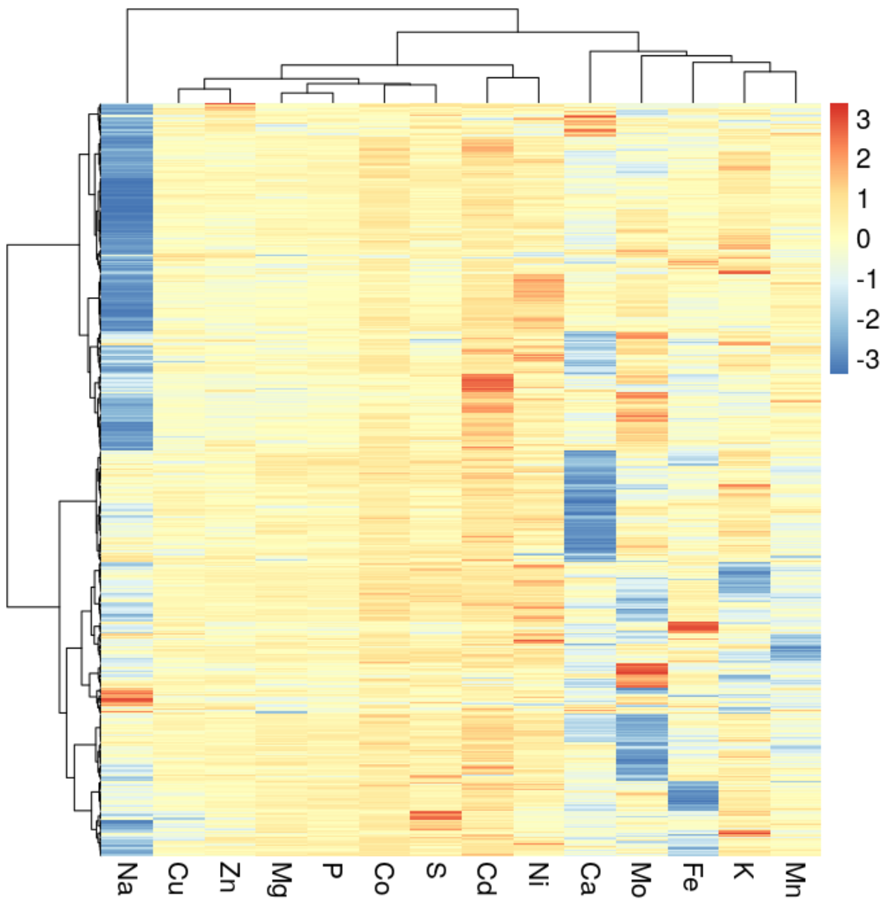

Figure 12: **Exploratory analysis plots with respect to ionome**

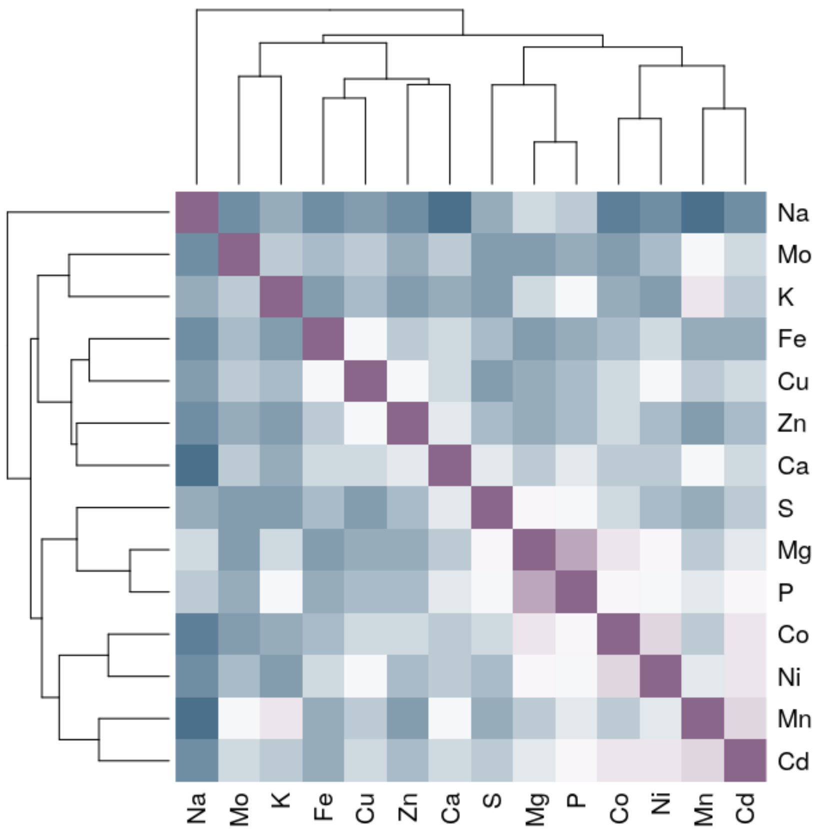

Figure 13: Exploratory analysis plots with respect to ionome

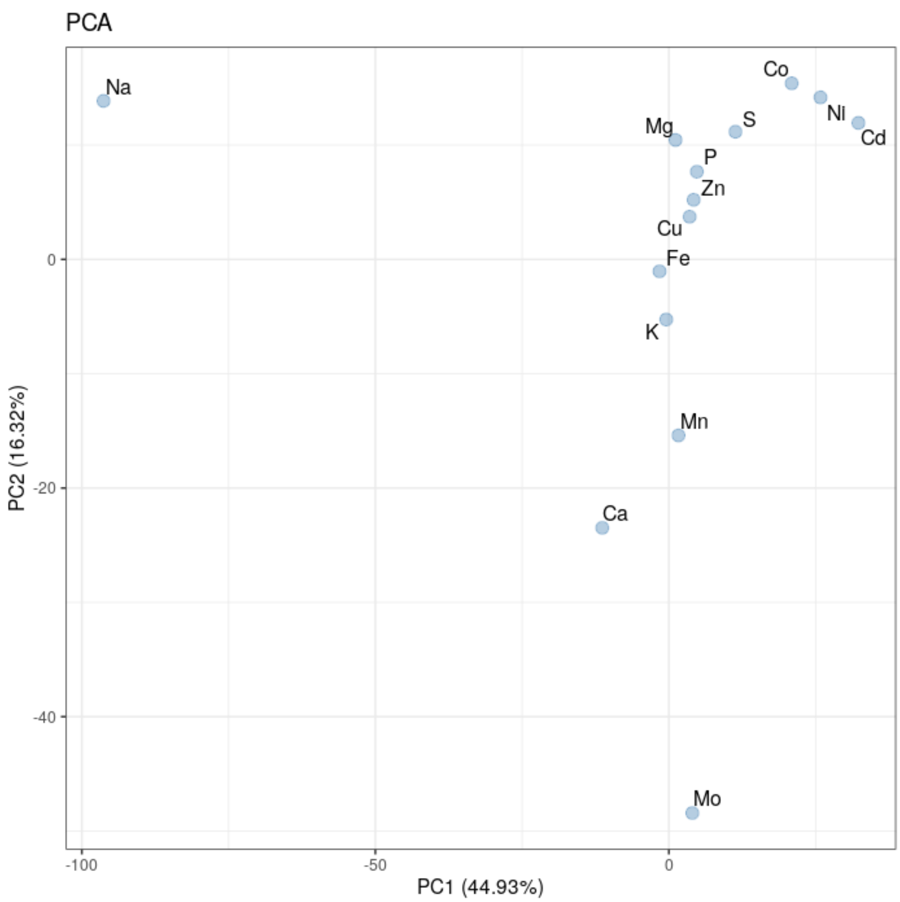

Figure 14: Exploratory analysis plots with respect to ionome

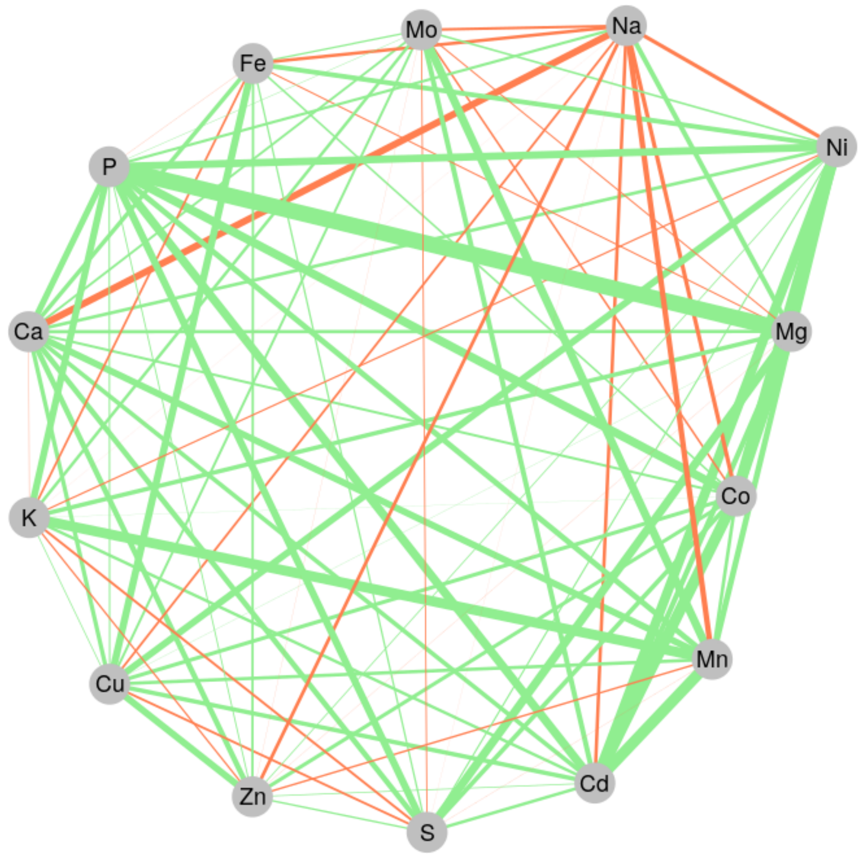

Figure 15: Exploratory analysis plots with respect to ionome