# EigenMS Examples

*Yuliya Karpievitch*

*3 October 2017*

## How to run EigenMS

The data used in the examaples here is a subset of a proteomics experiment where peptide IDs (sequences) have been shuffled and protein IDs were replced by fake 'prot_#' name. This document provides an example of the code and data structures that are necessary to run EigenMS. For non-proteomics data, such as metabolomics data, 2 columns with identical information can be provided.

```
ddata = read.table("test_peptides.txt", header=TRUE)
dim(ddata) # 1946 peptides by 4 samples
```

```
## [1] 1946    6
```

```
m_logInts = ddata[,3:6] # last 4 columns are the peptide intensities
head(m_logInts)
```

```
##         S1       S2       S3       S4
## 1 18.27488 18.13258 17.31412 16.65769
## 2 20.27289 20.23435 19.83706 20.40695
## 3 21.48269 20.72869 20.67476 19.94977
## 4 25.37097 24.51610 24.16154 24.09229
## 5 26.41770 25.40043 25.35071 24.84452
## 6 24.34607 23.66625 23.82716 22.55518
```

If peptide intensities are not on log scale and if there are 0's representing missing values, preform the following 2 lines. Fisrtreplace 0's with NA's as cannot take log2 of 0, in addition, 0 is a valid value and should not be in the data unless a peptide intensity is TRULY 0 and usually it is bellow the detection limit and not 0.

```
# m_logInts[m_logInts==0] = NA
# m_logInts = log2(m_logInts)
```

For discussion on why 0's should not be used in prteomics data analysis check out the following two publications:

*"Normalization and missing value imputation for label-free LC-MS analysis" Karpievitch YV, Dabney AR, Smith RD. BMC Bioinformatics 2012, PMID: 23176322*

*"Liquid Chromatography Mass Spectrometry-Based Proteomics: Biological and Technological Aspects" Karpievitch YV, Polpitiya AD, Anderson GA, Smith RD, Dabney AR. Ann Appl Statistics 2010, PMID: 21593992*

```
m_nummiss = sum(is.na(m_logInts)) # 1946 total values, 681 missing values
m_nummiss
```

```
## [1] 681
```

```
m_numtot = dim(m_logInts)[1] * dim(m_logInts)[2] # 8000 total observations
m_percmiss = m_nummiss/m_numtot  # 8.7% percent missing observations
m_percmiss
```

```
## [1] 0.08748715
```

```
# plot number of missing values for each sample
par(mfcol=c(1,1))
barplot(colSums(is.na(m_logInts)),main="Numbers of missing values in samples (group order)")
```

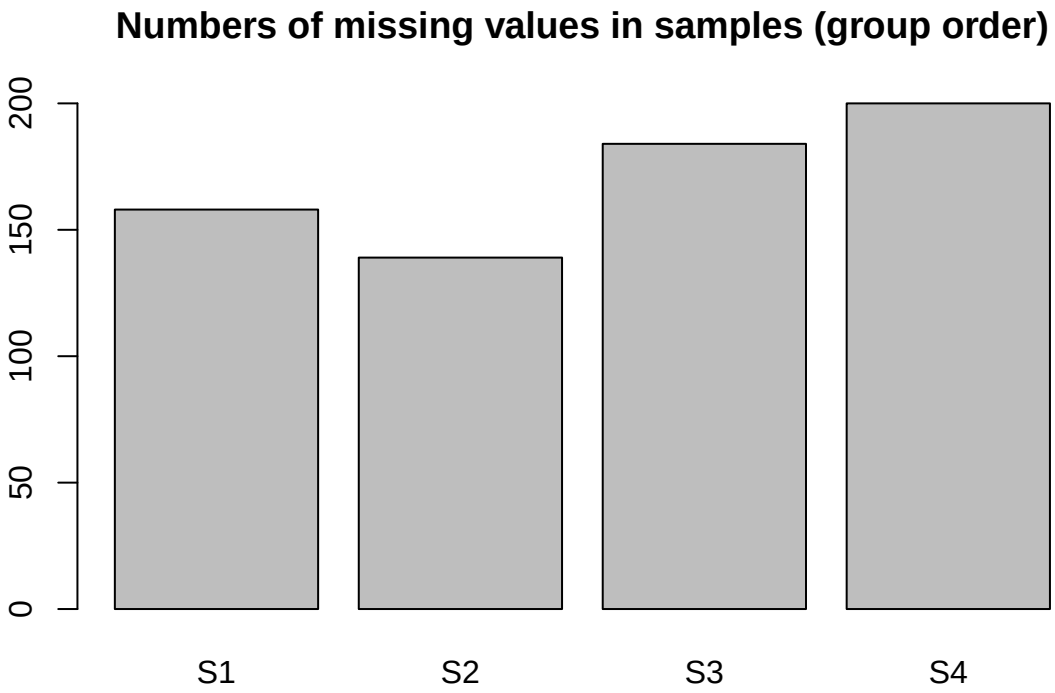



Figure 1. Numbers of missing values in each of the 4 samples.

Define parameter prot.info, 2 column data frame with IDs for metabolites or peptides in case of matabolites the 2 columns are identical. For peptides 1st column must contain unique peptide ID (usually sequences) 2nd column can contain protein IDs, but is not used in EigenMS, simply propagated back to the user to be used in further analyses.

```
m_prot.info = ddata[,1:2] # all unique metabolite/peptide IDs, otherwise not possible to have as row na
head(m_prot.info)
```

```
##                                pepID protID
## 1 AAAQTMRPNPAFSAEQVITELGVGEALISFLDEK prot_1
## 2                   AAAQVNMDLGLLPAER prot_2
## 3                 AAATGVMPLDMPESVLVR prot_3
## 4                   AADSHMAGFWEFPGGK prot_4
## 5                    AAELEVVLPLSFFEK prot_5
## 6                       AAETIDVSLPGR prot_6
```

**Example 1 - single factor normalization with 2 treatment groups**

```
grps = as.factor(c(1,2,1,2)) # 1 = control; 2 = treatment
# note that treatment groups variable must be a 'factor'
m_ints_eig1 = eig_norm1(m=m_logInts,treatment=grps,prot.info=m_prot.info)
```

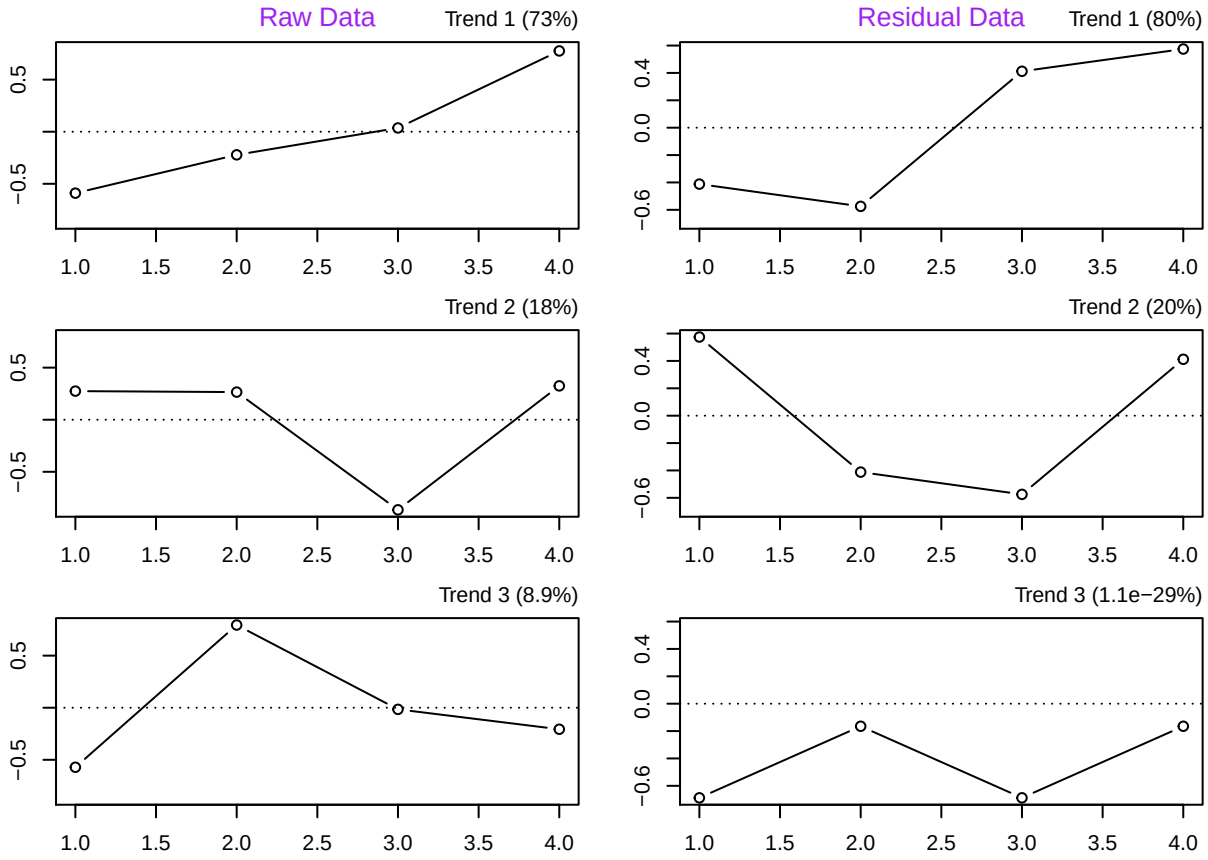



Figure 2. Eigetrends for raw and residual peptide intensities. Dots at positions 1, 2, 3 and 4 correspond to the 4 samples: S1, S2, S3 and S4 in that order. Top trend in the Residual Data shows that sample S1 and S2 have high similarity, as well as, S3 and S4 whereas in reality samples S1 and S3 are from the same treatment group and S2 and S4 are from the other.

```
m_ints_eig1$h.c # 1 bias trend estimated
m_ints_norm1 = eig_norm2(rv=m_ints_eig1)
```

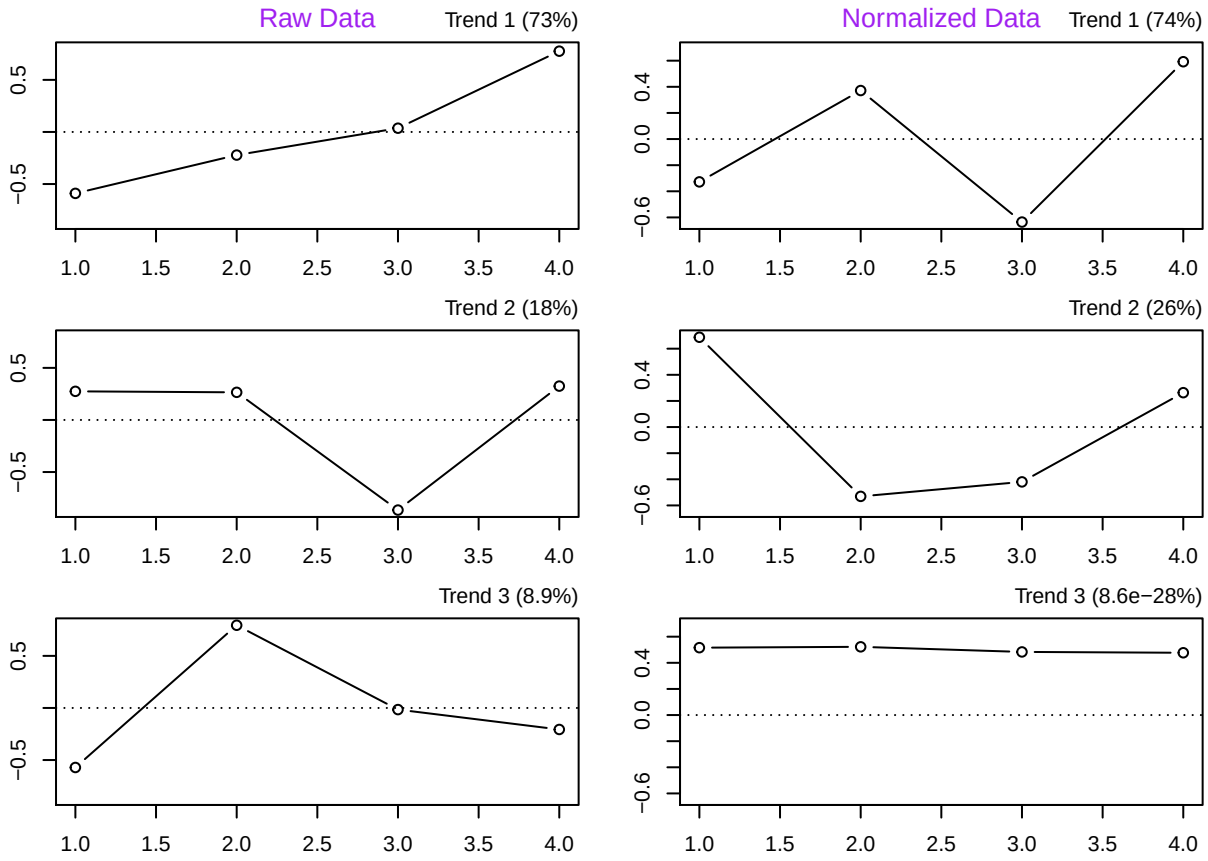



Figure 3. Eigetrends for raw and residual peptide intensities. Dots at positions 1, 2, 3 and 4 correspond to the 4 samples: S1, S2, S3 and S4 in that order. Top trend in the Normalized Data shows that sample S1 and S3 have high similarity, as well as, S2 and S4 as expected for biological replicated in teh same treatment group.

```
par(mfcol=c(1,2))
boxplot(m_logInts, las=2, main='Raw intensities')
boxplot(m_ints_norm1$norm_m, las=2, main='Normalized intensities')
```

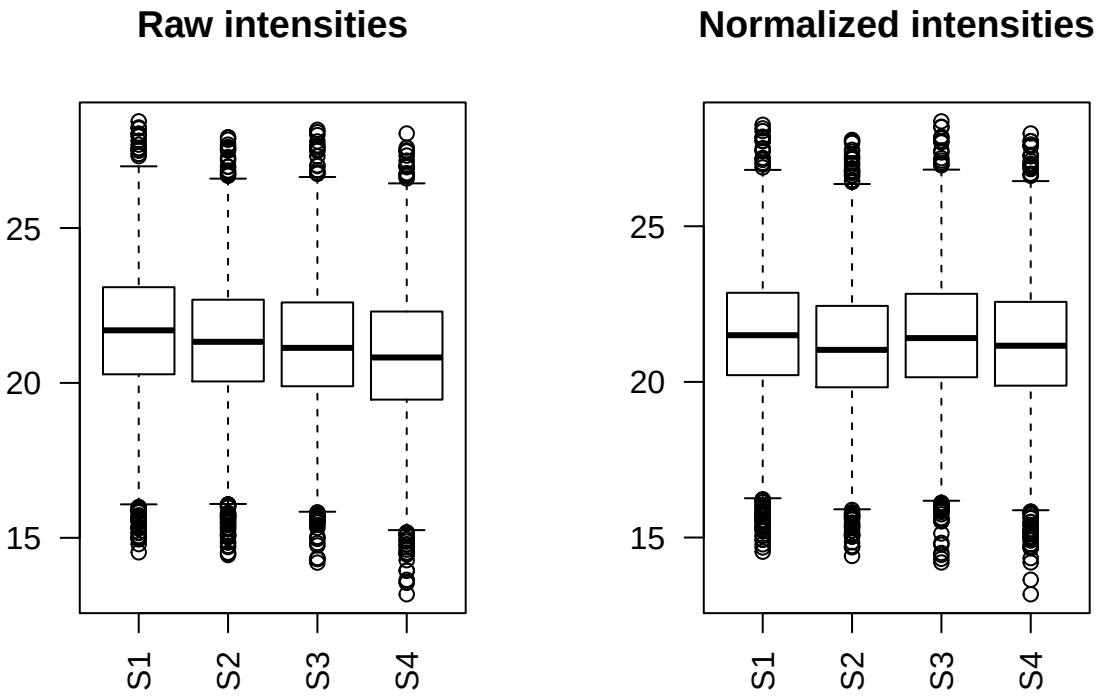



Figure 4. Raw (left panel) and normalized (right panel) peptide intensities.

Raw intensities have a decreasing intensity from S1 to S4 showing potential LC column clogging effect. Normalized data shows the biologic differences previously obscured by the decreasing intensity over time where samples S1 and S3 have higher intensities and smaples S2 and S4 have lower intensities.

```
heatmap.2(as.matrix(m_logInts), dendrogram='none', Rowv = NULL, Colv = NULL, trace='none')
```

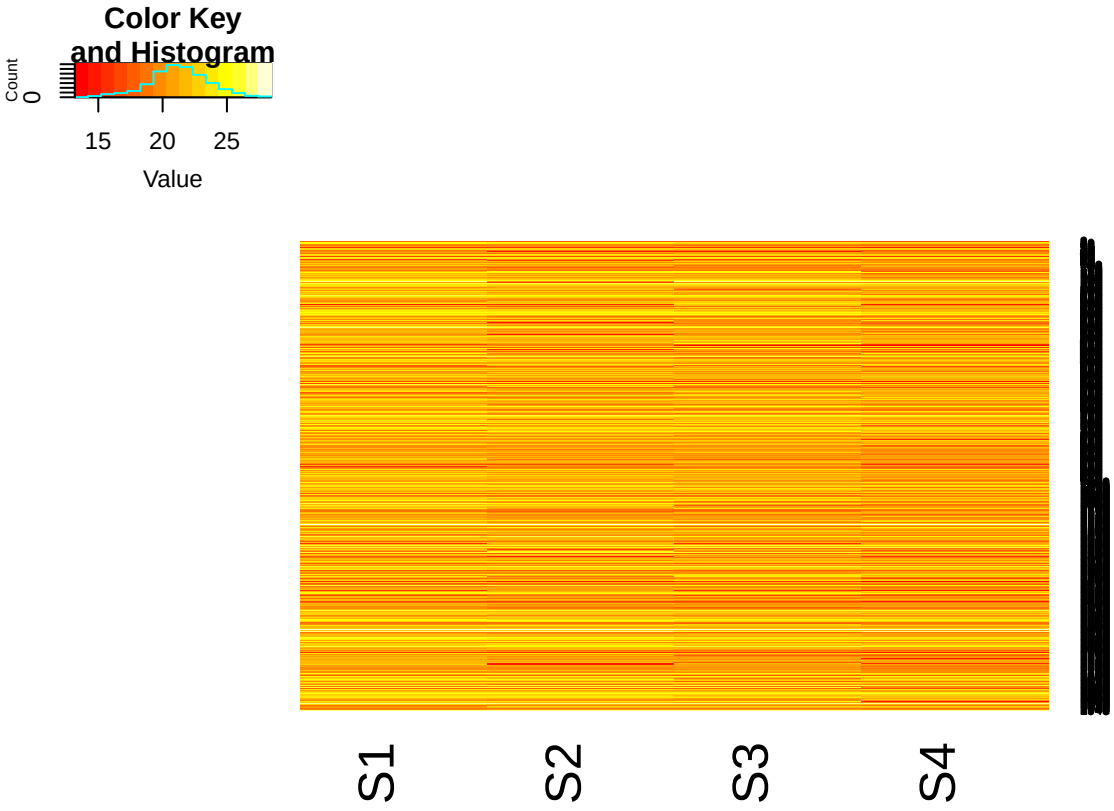


Figure 5. Raw peptide intensities.

```
heatmap.2(m_ints_norm1$norm_m, dendrogram='none', Rowv = NULL, Colv = NULL, trace='none')
```

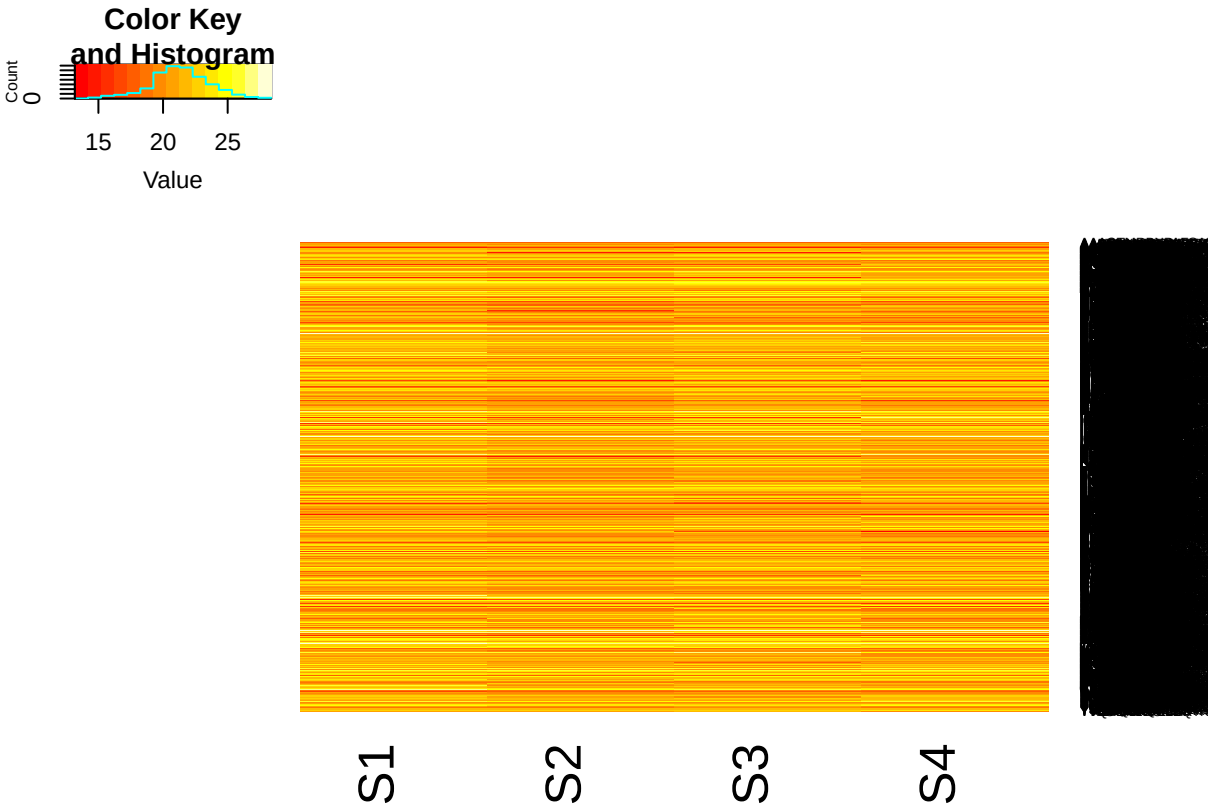
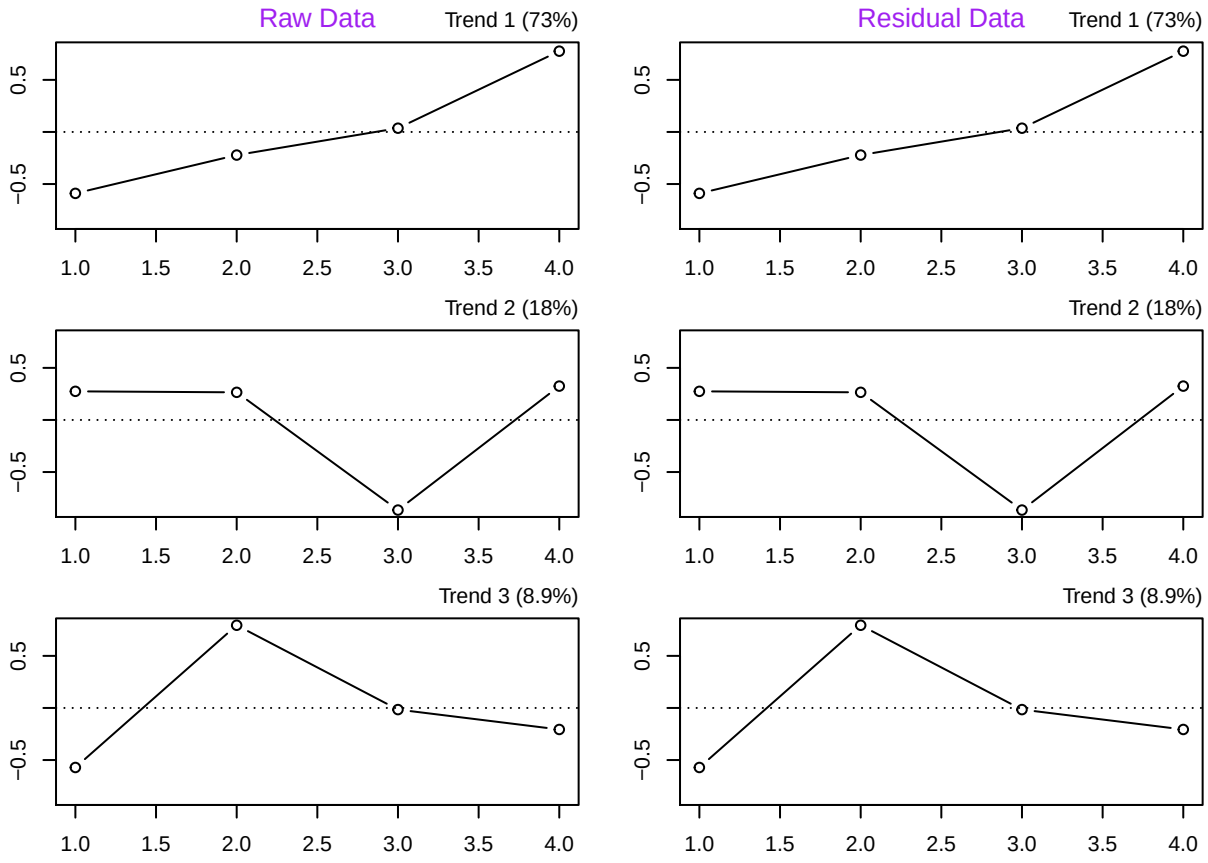



Figure 6. Normalized (bottom panel) peptide intensities.

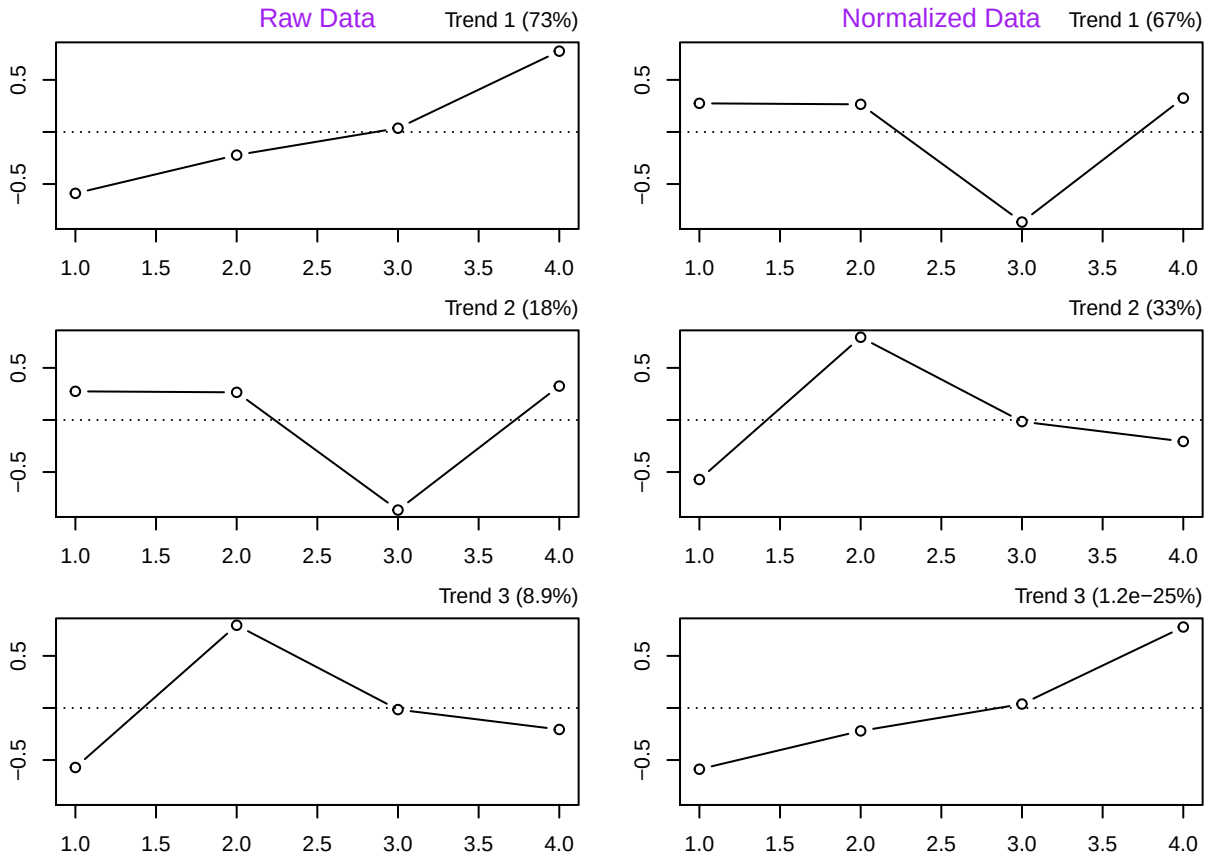
**Example 2 - single factor normalization with 1 treatment groups**

Here techinical as well as biological variation will be normalized out.

```
grps2 = as.factor(c(1,1,1,1))
m_ints_eig1_v2 = eig_norm1(m=m_logInts,treatment=grps2,prot.info=m_prot.info)
```




```
m_ints_norm1_v2 = eig_norm2(rv=m_ints_eig1_v2)
```




```
par(mfcol=c(1,2))
boxplot(m_logInts, las=2, main='Raw intensities')
boxplot(m_ints_norm1$norm_m, las=2, main='Normalized intensities')
```

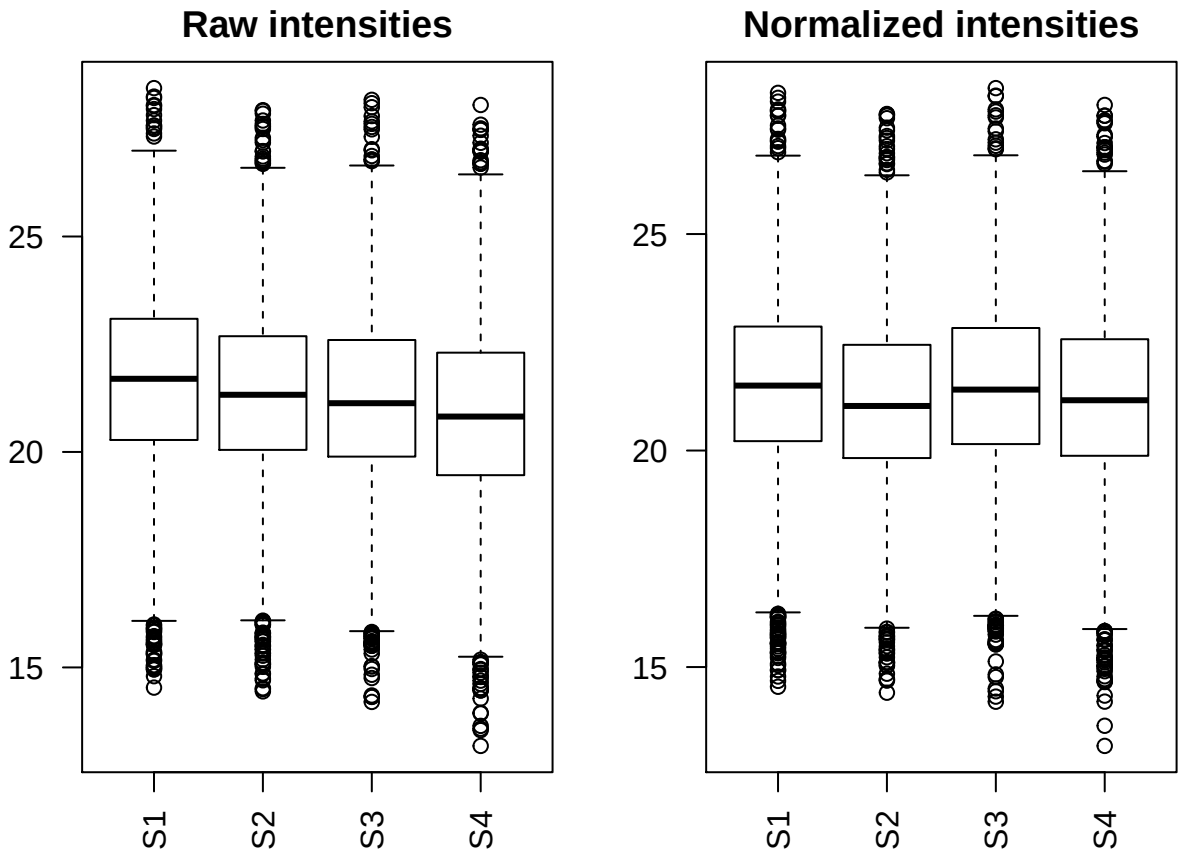



Figure 7. Raw (left panel) and normalized (right panel) peptide intensities. Normalization was performed with a single treamtent group thus reducing most of the variation present in the data, techinical and biological. *Avoid doing this on data where biological variation is of interets.

**Example 3 - nested treatment groups - simulated larger dataset**

Make matrix of data 2 times larger by copying existing columns for the simulated treatment group 1 and using the same colums with added variation (uniform(0, .3)) Thus, most intensities are slightly higher for the treatment group 2.

```
m_logInts2 = cbind(m_logInts, m_logInts+runif(dim(m_logInts)[1], 0, .3))
gr1 = as.factor(c(1,2,1,2,1,2,1,2)) # similar to S1, S2, ... in biological data
gr2 = as.factor(c(1,1,1,1,2,2,2,2))
grps2 = data.frame(gr1,gr2)

m_ints_eig2 = eig_norm1(m=m_logInts2,treatment=grps2,prot.info=m_prot.info)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     gr1, gr2
```

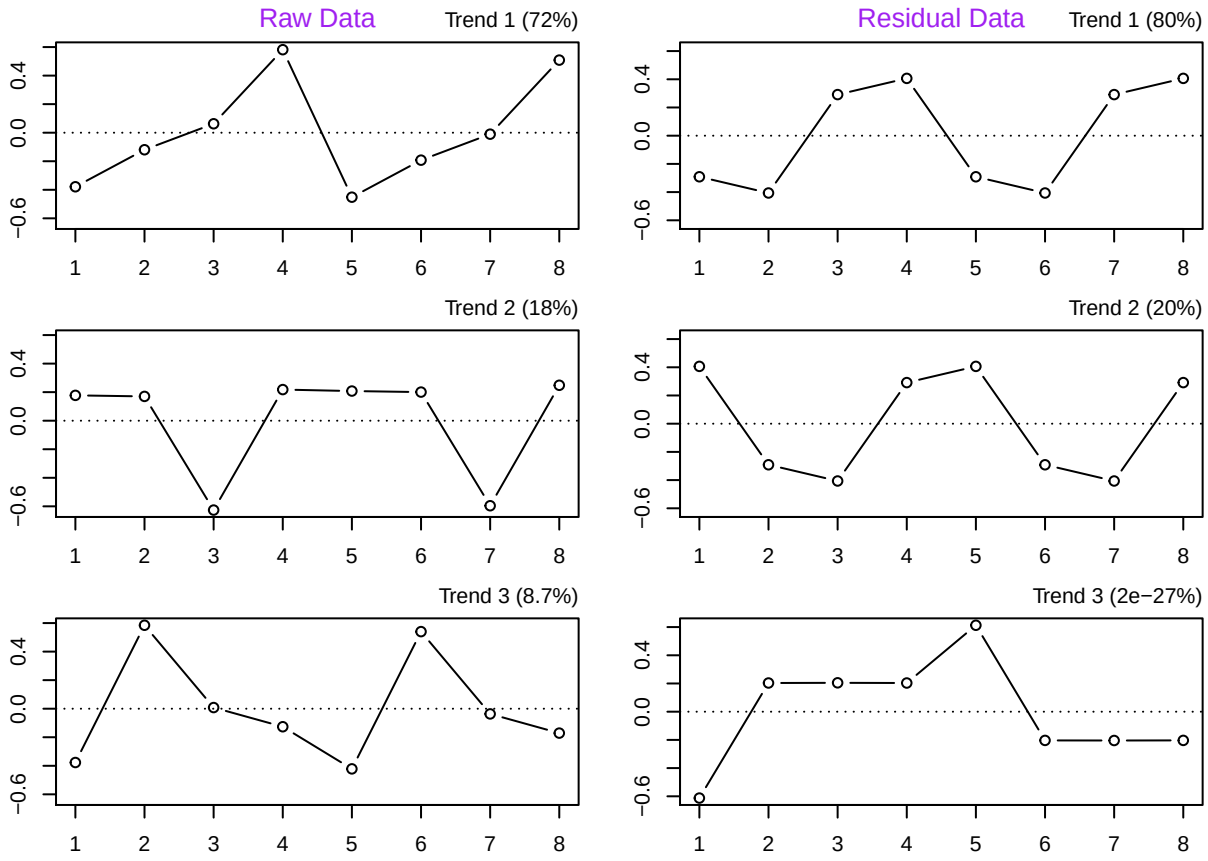



Figure 8. Eigetrends for raw and residual peptide intensities. Dots at positions 1-84 correspond to the 8 samples: S1-S8 in that order.

```
m_ints_norm2 = eig_norm2(rv=m_ints_eig2)
```

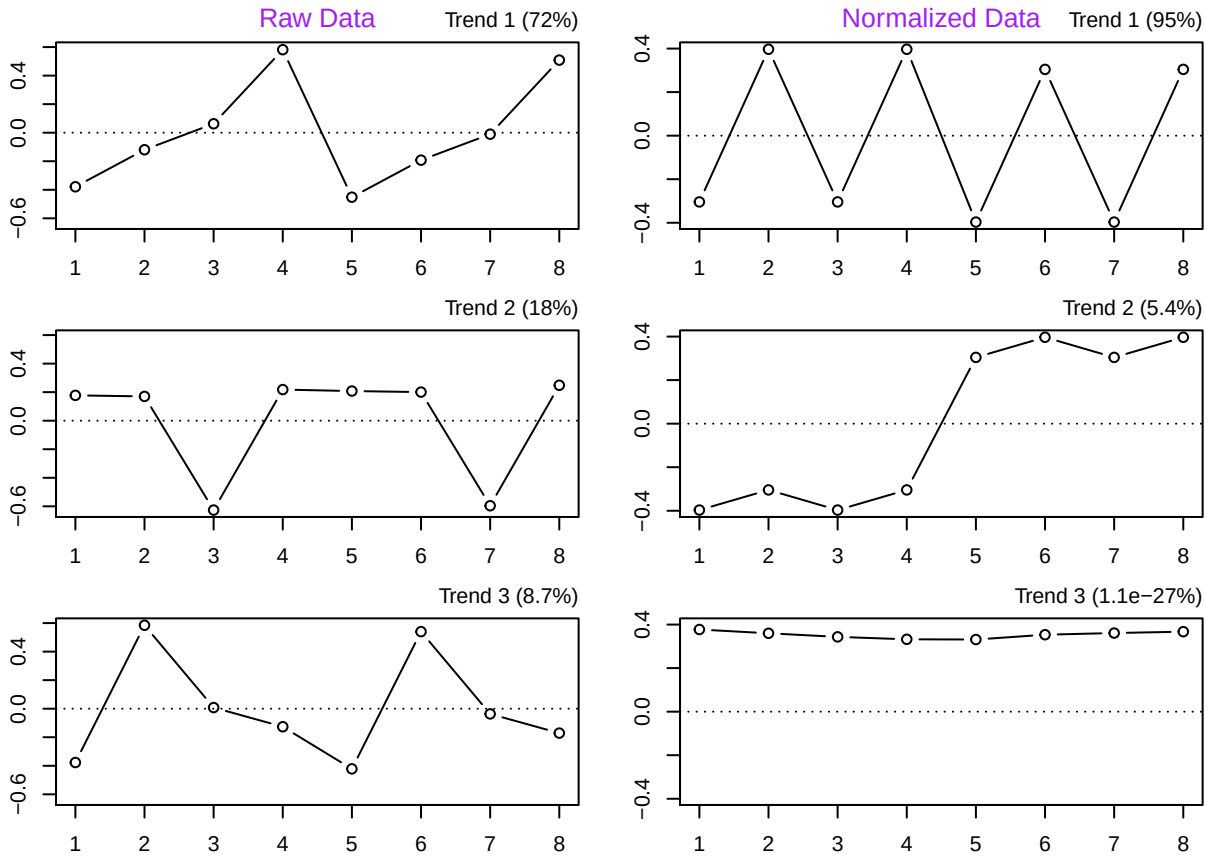



Figure 9. Eigetrends for raw and normalized peptide intensities. Dots at positions 1-8 correspond to the 8 samples: S1-S8 in that order. Top trend in the Normalized Data shows that sample S1, S3, S5 & S7 have high similarity, as well as, S2, S4, S6 & S8 as expected for biological replicated in the same treatment group. In addition 2nd eigentrend in Normalized Data panel shows the pattern that we introduced as gr2 (first 4 samples lower in intensity than the last 4).

EigenMS normalization had preserved the two variations of interest encoded by gr1 and gr2 treatments while removing the descrease in intensity that is present in the raw data (top panel in Raw Data). Note that eigentrends can be rotated arnd the x-axis thus the top trend represents intensity loss from smaple S1 to S4, then again from sample S5 to S8 simulating LC column replacement mid-experiment.

```
par(mar=c(10,4,4,4)) # allows to have nice vertical labels for longer labels
par(mfcol=c(1,2))
boxplot(m_logInts2, las=2) # raw data
boxplot(m_ints_norm2$norm_m, las=2)  # normalized data
```

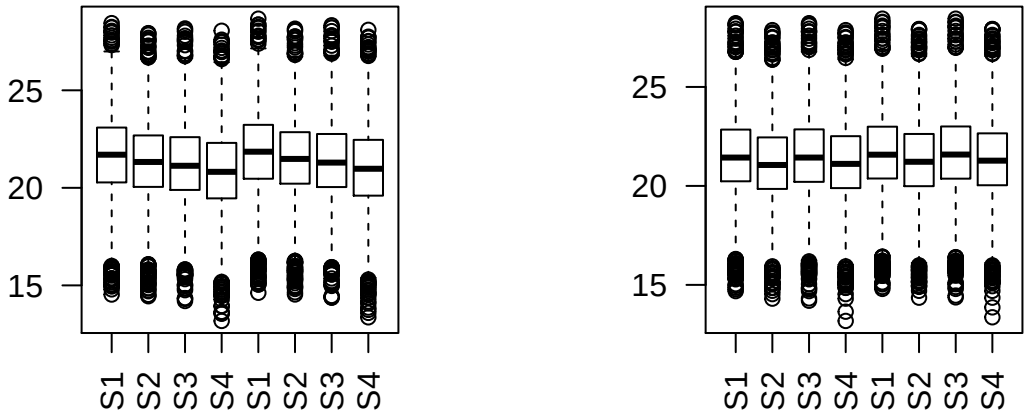



Figure 10. Raw (left panel) and normalized (right panel) peptide intensities for simulated dataset. Normalized data shows the expected zig-zag pattern for the treatment groupsencoded by gr1 which explains most of teh variation in the data.