Multiomics in cancer

Jieun Jeong

2025-03-22

The most typical types of NGS data collected from cancer cell cultures and biopsies are mRNAseq for gene expression, copy number variations of genes and mutation status of genes.

Moreover, there are metadata like cancer subtypes that are determined by other methods, like morphology of cells in the biopsy or patient symptoms.

All those data can be analyzed together with aims like target identifications or markers for precision medicine.

We first download data and create a data frame of selected metadata.

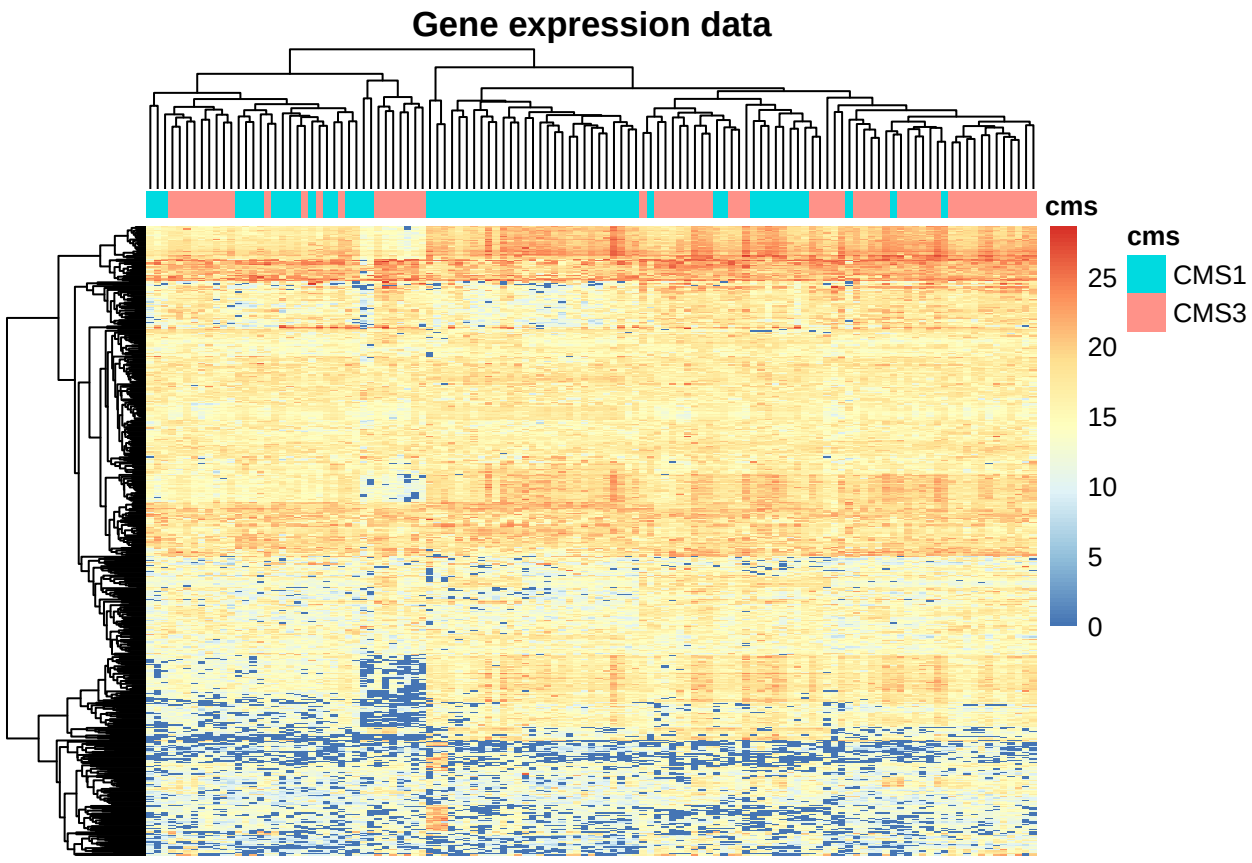
```
# input prefix
paD <- "/Users/jieun/Work/Multiomics/Datasets/COREAD_CMS13_"
DataTable <- function(x) read.csv(paste0(paD,x,".csv"), row.names=1,
                          check.names = FALSE)
DataMatrix <- function(x) as.matrix(DataTable(x))
# column: patient, row: gene or cytoband
x1 <- DataMatrix("gex")  # gene expression
x2 <- DataMatrix("muts") # mutation data
x2[x2 > 0] = 1            # set to binary
x3 <- DataMatrix("cnv")  # copy number variation
# columns: metadata types, rows: patients
covariates <- DataTable("subtypes")
covariates <- covariates[colnames(x1),] # assure match with data tables
anno_col <- data.frame(cms=as.factor(covariates$cms_label),
                       row.names = rownames(covariates))
```

Heatmaps

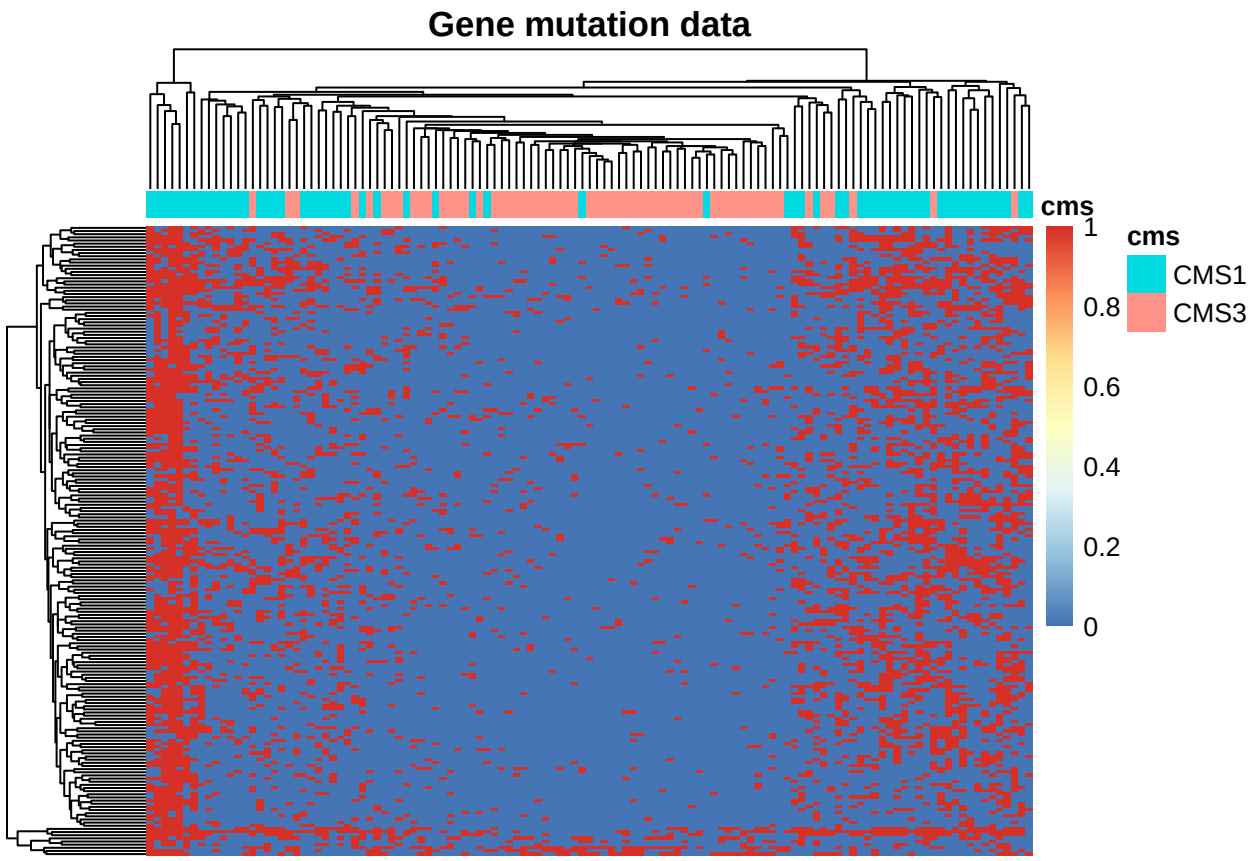
The most straighforward way of checking the connection between data and cancer subtypes is plotting heatmaps.

```
Heatmap <- function(x,y,z=FALSE) pheatmap::pheatmap(x, annotation_col = anno_col,
              show_colnames = FALSE, show_rownames = z, main=y)
```

```
Heatmap(x1,"Gene expression data")
```

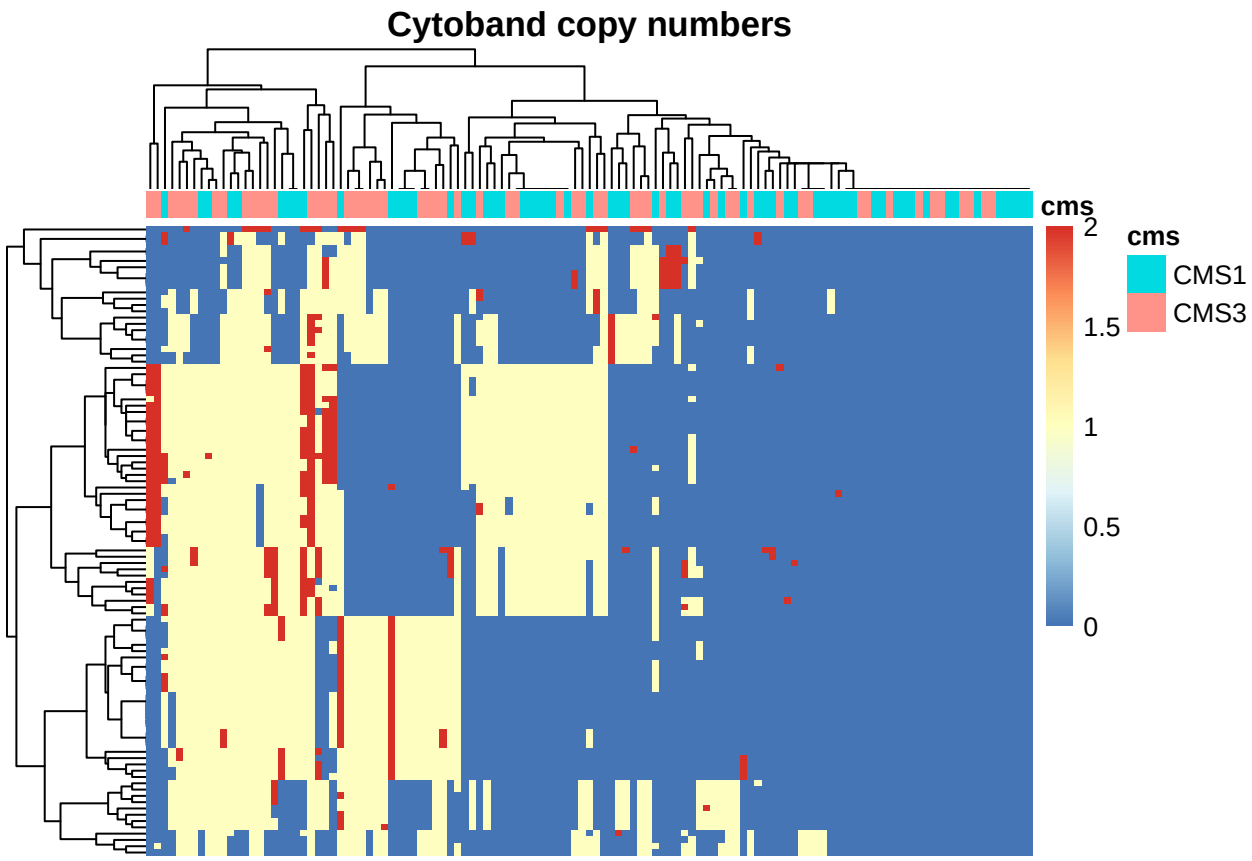



```
Heatmap(x2,"Gene mutation data")
```



Copy number variation data was normalized in such a way that for each cytoband row there are only three values among 121 patient entries: $0 < S < L$ with $L = 2S$, so we transform it to emphasize this fact:

```
Heatmap(((x3 > 0.1) + (x3 > 0.2)),"Cytoband copy numbers")
```




```
# consider ComplexHeatmap
```

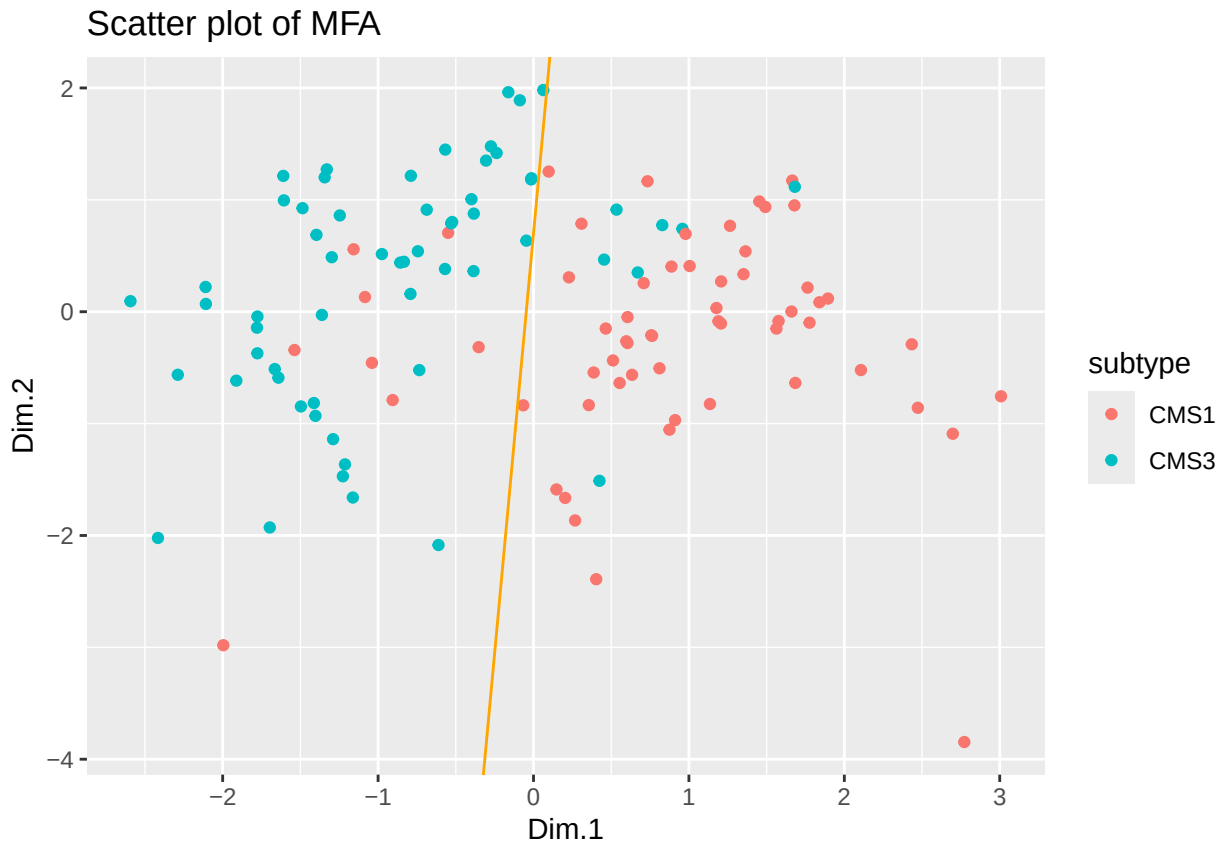
Latent variable models for multi-omics integration

While there are many aspects to consider in multiomics, like feature selections for every -omic, normalization, trying composite variables like "gene X loss of function" that can be reflected in mutation, low copy number or low expression, but the general challenge is dimenension reduction that is appropropriate for hererogenous data.

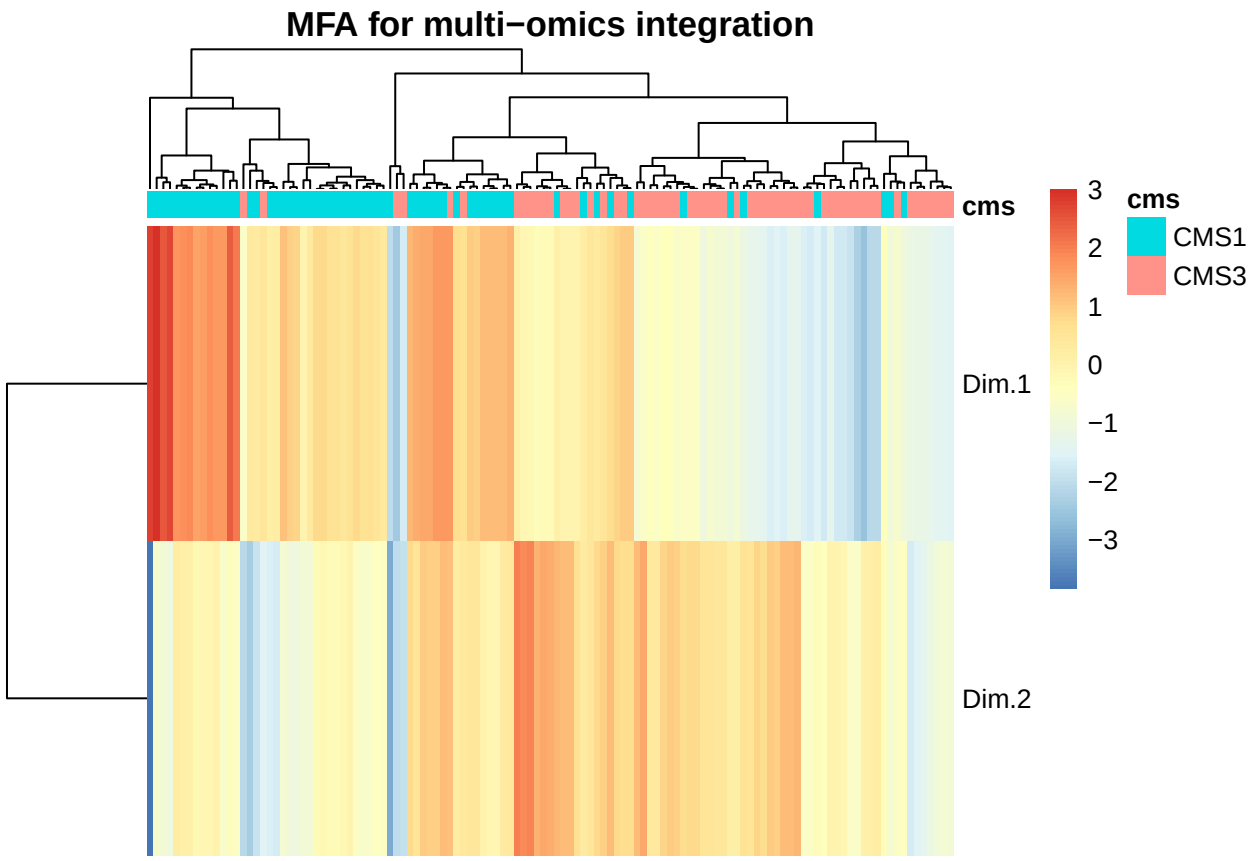
Multiple factor analysis

```
 r.mfa <- FactoMineR::MFA(
  t(rbind(x1,x2,x3)), # binding the omics types together
  c(dim(x1)[1], dim(x2)[1], dim(x3)[1]), # specifying the dimensions of each
  graph = FALSE)

# first, extract the H and W matrices from the MFA run result
mfa.h <- r.mfa$global.pca$ind$coord
mfa.w <- r.mfa$quanti.var$coord

# create a data frame with the H matrix and the CMS label
mfa_df <- as.data.frame(mfa.h)
mfa_df$subtype <- factor(covariates[rownames(mfa_df),]$cms_label)
```

```r
# create the plot
library(ggplot2)
ggplot(mfa_df, aes(x=Dim.1, y=Dim.2, color=subtype)) +
  geom_point() + ggtitle("Scatter plot of MFA") +
  geom_abline(intercept = 0.7,slope = 15, color = "orange")
```




```r
pheatmap::pheatmap(t(mfa.h)[1:2,], annotation_col = anno_col,
                   show_colnames = FALSE,
                   main = "MFA for multi-omics integration")
```



We have 61 CMS1 cases and 60 for CMS3, prediction "if Dim.1 > 0 then CMS1" makes 8 false positives and 9 false negative. Using a linear function with small coefficient for Dim.2 we can reduce both false negatives and false positives by 1. Further optimization may be possible if we apply additional dimensions.

Non-negative matrix factorization, NMF

In this alternative approach to dimension reduction we require input matrix to be non-negative, and the same for matrices in the decomposition.

Normalization steps:

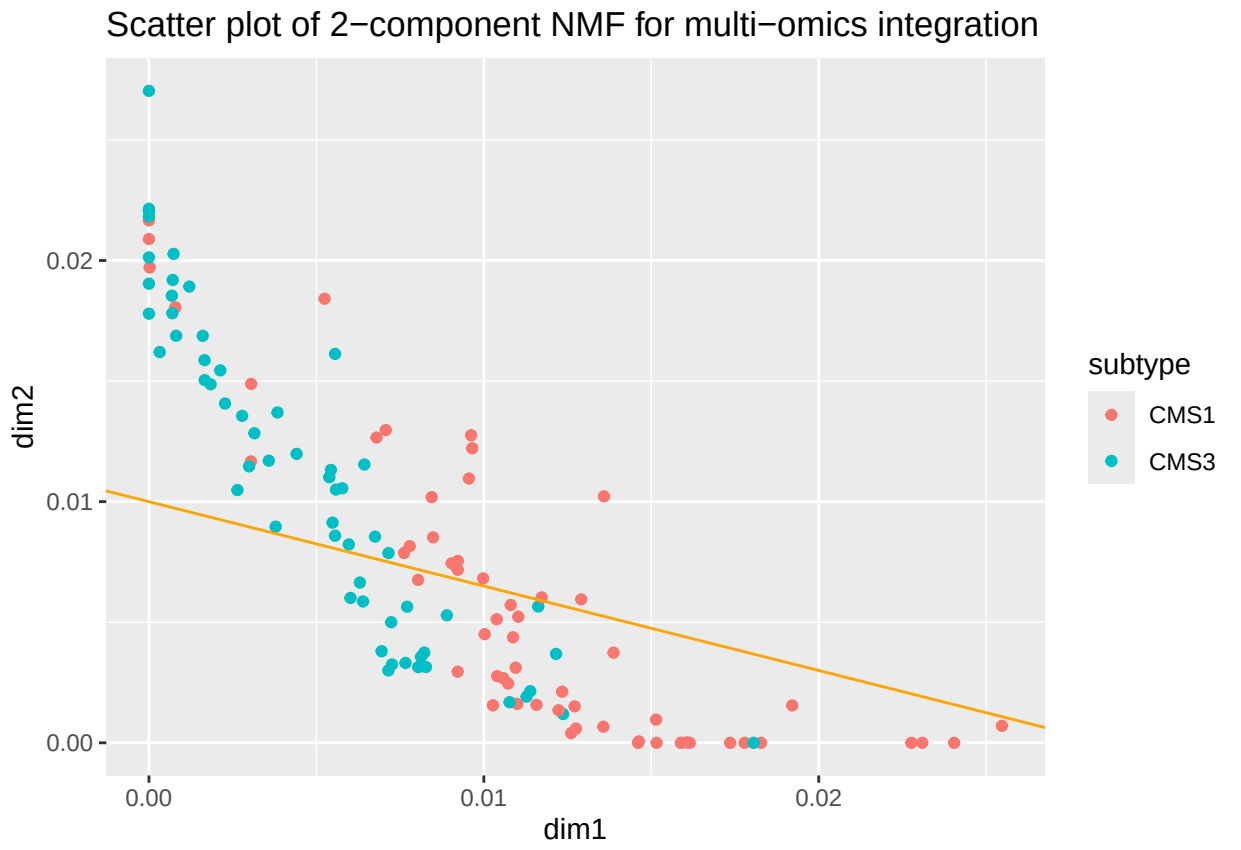
```
# Feature-normalize the data
x1.featnorm <- x1 / rowSums(x1)
x2.featnorm <- x2 / rowSums(x2)
x3.featnorm <- x3 / rowSums(x3)

# Normalize by each omics type's frobenius norm
x1.featnorm.frobnorm <- x1.featnorm / norm(as.matrix(x1.featnorm), type = "F")
x2.featnorm.frobnorm <- x2.featnorm / norm(as.matrix(x2.featnorm), type = "F")
x3.featnorm.frobnorm <- x3.featnorm / norm(as.matrix(x3.featnorm), type = "F")
```

Elimination of negative values

```
split_neg_columns <- function(x) {
  if (all(x >= 0)) return(x)
  # add code, not needed in our data
}
```

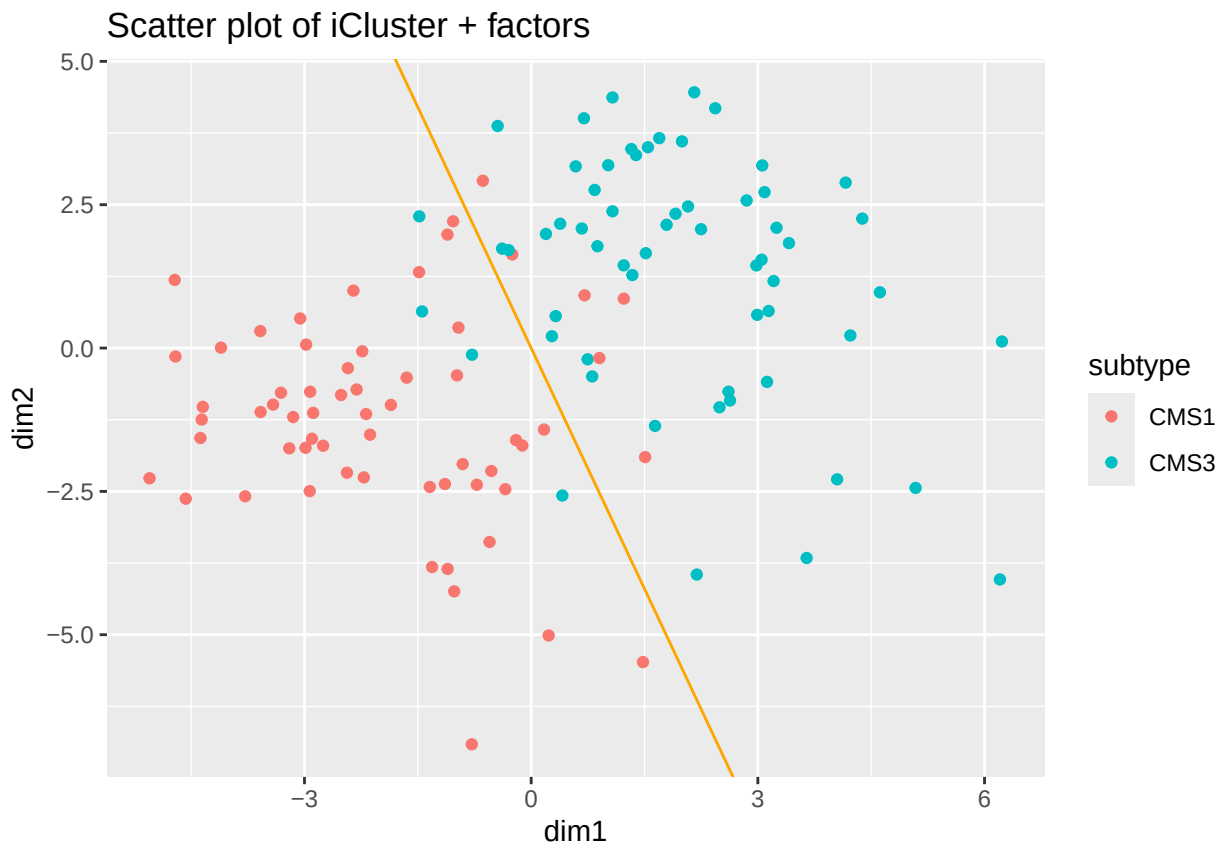
The application of NMF

```
require(NMF)
r.nmf <- NMF::nmf(t(rbind(x1.featnorm.frobnorm,
                          x2.featnorm.frobnorm,
                          x3.featnorm.frobnorm)),
                  2,
                  method = 'Frobenius')

# extract the H and W matrices from the nmf run result
nmf.h <- NMF::basis(r.nmf)
nmf.w <- NMF::coef(r.nmf)
nmfw <- t(nmf.w)

# create a dataframe with the H matrix and the CMS label (subtype)
nmf_df <- as.data.frame(nmf.h)
colnames(nmf_df) <- c("dim1", "dim2")
nmf_df$subtype <- factor(covariates[rownames(nmf_df),]$cms_label)

# create the scatter plot
ggplot(nmf_df, aes(x = dim1, y = dim2, color = subtype)) + geom_point(size=1.5) +
   geom_abline(slope=-0.35,intercept=0.01,color="orange") +
  ggtitle("Scatter plot of 2-component NMF for multi-omics integration")
```

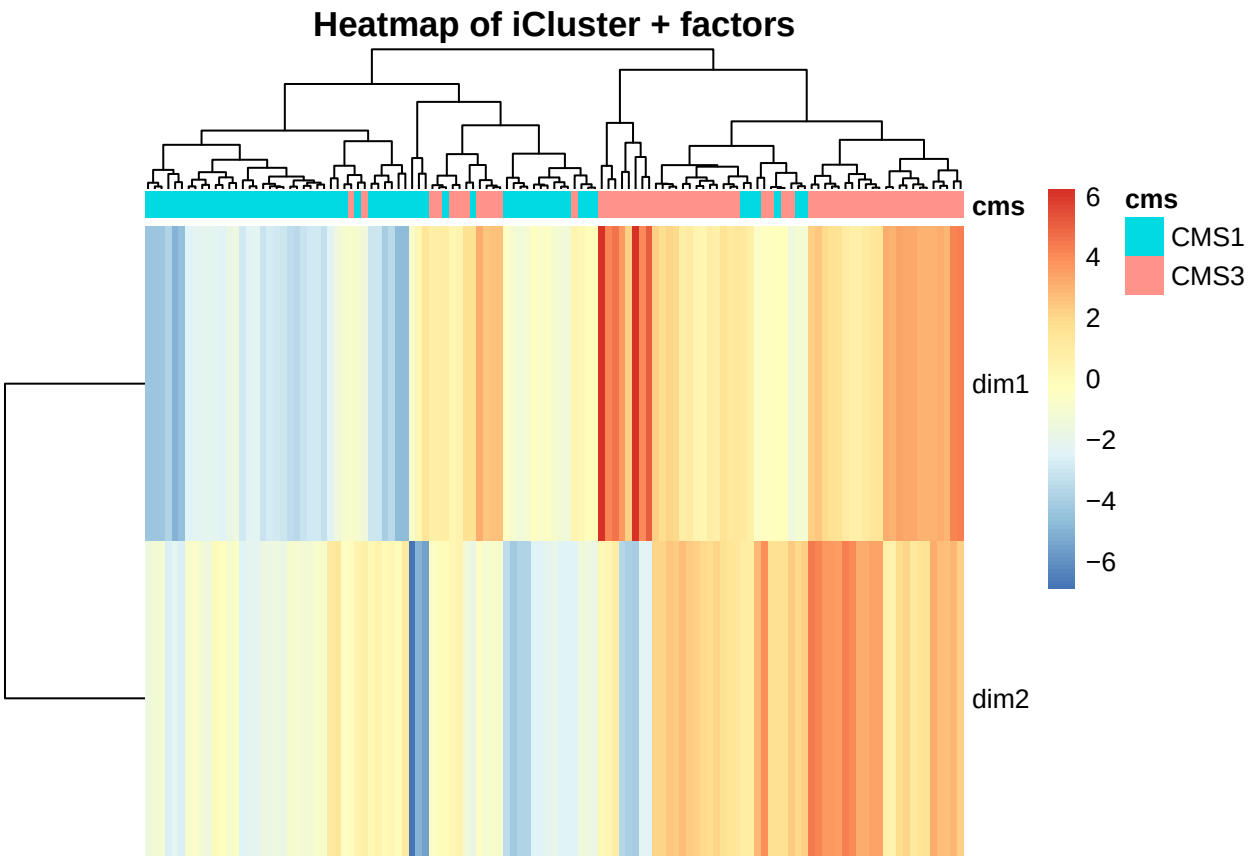


In this case, dim1 + dim2 from NMF do not better in separating CMS1 from CMS2, but in another context NMF can be considered. ## Clustering This method seems best for our data.

```r
require(iClusterPlus)

# run the iClusterPlus function
r.icluster <- iClusterPlus::iClusterPlus(
  t(x1), # Providing each omics type
  t(x2),
  t(x3),
  type = c("gaussian", "binomial", "multinomial"), # Providing the distributions
  K = 2, # Provide the number of factors to learn
  alpha = c(1, 1, 1), # as well as other model parameters
  lambda = c(.03, .03, .03)
)

# extract the H and W matrices from the run result
# here, I refer to H as z, to keep with iCluster terminology
icluster.z <- r.icluster$meanZ
rownames(icluster.z) <- rownames(covariates) # fix the row names
icluster.ws <- r.icluster$beta

# construct a data frame with the H matrix (z) and the cancer subtypes
# for later plotting
icp_df <- as.data.frame(icluster.z)
colnames(icp_df) <- c("dim1", "dim2")
rownames(icp_df) <- colnames(x1)
icp_df$subtype <- factor(covariates[rownames(icp_df),]$cms_label)


# create the plot
ggplot(icp_df, aes(x=dim1, y=dim2, color=subtype)) + geom_point() +
  ggplot2::ggtitle("Scatter plot of iCluster + factors") +
  geom_abline(slope=-2.8,color="orange")
```




```
pheatmap::pheatmap(t(icp_df[,1:2]), annotation_col = anno_col,
                   show_colnames = FALSE,
                   main = "Heatmap of iCluster + factors")
```


Heatmap of iCluster + factors
cms
dim1
dim2
6
4
2
0
−2
−4
−6
cms
CMS1
CMS3