PAC1_Genòmica computacional

Julia Baguña Torres

2025-03-31

Taula de continguts

Abstract

Aquest treball s'ha centrat en l'anàlisi d'un conjunt de dades de fosfoproteòmica per tal d'identificar la presència de modificacions posttraduccionals de fosfopèptids a través de l'espectrometria de masses en mostres tumorals. S'ha utilitzat un enfocament de reducció de dimensionalitat del dataset mitjançant l'anàlisi de components principals (PCA) per determinar si existia una separació de mostres tumorals en funció del fenotip de fosfoproteòmica i identificar possibles patrons diferencials entre grups. A través d'un anàlisi de clustering k-means, s'han identificat tres subgrups de mostres tumorals. A més a més, s'han determinat els fosfopèptids més influents sobre cadascun dels component principals identificats, proporcionant informació sobre la seva contribució sobre la separació entre les mostres. L'anàlisi confirma que la fosforilació i altres modificacions posttraduccionals podrien tenir un paper clau en la diferenciació de fenotips tumorals, oferint noves vies per a l'exploració de biomarcadors en càncer.

Objectius

L'objectiu principal d'aquest estudi és analitzar un conjunt de dades de fosfoproteòmica de 1438 fosfopèptids en mostres tumorals PDX mitjançant un enfocament bioinformàtic. Es pretén:

- Determinar si existeixen agrupaments clars entre les mostres tumorals
- Identificar els fosfopèptids clau que contribueixen a la separació entre subgrups de mostres.

Mètodes

Descàrrega de dades i preparació del dataset:

El conjunt de dades s'ha descarregat des del repositori GitHub (https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2018-Phosphoproteomics (https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2018-Phosphoproteomics)) en format .xlsx, utilitzant la llibreria readxl per a la lectura de l'arxiu. Després, s'ha organitzat en una matriu de dades quantitatives i una taula de metadades. L'objecte SummarizedExperiment s'ha generat per emmagatzemar les dades i les metadades associades als fosfopèptids i mostres.S'ha realitzat un filtratge per eliminar files de l'objecte SummarizedExperiment amb variància zero.

Anàlisi exploratòria i PCA:

S'han utilitzat gràfics de boxplot per observar la distribució de les intensitats dels fosfopèptids entre les diferents mostres. Per reduir la dimensionalitat del dataset i identificar els components principals, s'ha dut a terme un PCA sobre les dades, utilitzant les 12 mostres per identificar la variància explicada per cada component principal. S'ha generat un gràfic de columnes per visualitzar la proporció de variància explicada per

cada component. Posteriorment, s'ha comprovat la correlació entre els components PC1, PC2 i PC3, i s'ha realitzat un gràfic de dispersió en 2D i un gràfic de dispersió en 3D mitjançant les llibreries ggplot i plotly,respectivament, per visualitzar les separacions de les mostres en funció d'aquests components.

Anàlisi de clustering K-means:

S'ha aplicat l'algorisme de clustering K-means per identificar subgrups entre les mostres. Aquest mètode ha validat la separació observada a través del PCA, identificant tres clústers de mostres.

Identificació dels fosfopèptids més influents: A partir dels "loadings" del PCA, s'han identificat els fosfopèptids més influents en la separació dels components principal. Aquests fosfopèptids han estat associats amb les modificacions de seqüència per a determinar la seva rellevància biològica.

Resultats

He descarregat el dataset 2018-Phosphoproteomics en format .xlsx des de l'enllaç del repositori Github proporcionat.

https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2018-Phosphoproteomics (https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2018-Phosphoproteomics)

He descarregat el paquet readxl i Biocmanager per treballar amb aquest tipus d'arxius i carregat les respectives llibreries amb les instruccions:

Creació de l'objecte de classe SummarizedExperiment:

A l'arxiu descarregat hi han les dades de les senyals d'abundància normalitzada d'espectrometria de masses de 1438 fosfopèptids a 6 mostres de PDX, amb duplicats tècnics de cadascuna (12 en total). Al dataset trobem les següents columnes:

SequenceModifications: Modificacions en les seqüències dels fosfopèptids

Accession: Identificador únics per a cada fosfopèptid

Description: Descripció detallada de cada fosfopèptid

Score: Valor associat a la qualitat de la mesura

Mostres: Les mostres individuals que han estat analitzades (12 mostres en total)

CLASS: Categoria del fosfopèptid

PHOSPHO: Presència de fosforilació o altres modificacions

Per tal de generar un objecte de classe SummarizedExperiment que contingui tant les dades quantificades (assay) com les metadades associades a cada metabolit i mostra he seguit els següents passos:

Primer de tot, he carregat l'arxiu d'excel des de la seva ubicació:

```
data <- read_excel("E:/Màster Bioinformàtica + bioestadística - UOC/5th semester/Genòmica com
putacional/PAC1/TIO2+PTYR-human-MSS+MSIvsPD.xlsx")
```

Tot seguit, he emmagatzemat les dades quantitatives de les 12 mostres (columnes 5 a 16 del conjunt de dades) en una matriu assay_data mitjançant la instrucció:

```
assay_data <- as.matrix(data[, 5:16])
```

Després, he creat un dataframe anomenat row_data que englobi les metadades associades als metabolits (la modificació de seqüència, el codi d'accés, la descripció, el score, la classe i la presència de fosforilació):

```
row_data <- DataFrame( SequenceModifications =
data$SequenceModifications, Accession = data$Accession, Description =
data$Description, Score = data$Score, CLASS =
data$CLASS, PHOSPHO = data$PHOSPHO)
```

De la mateixa manera, he creat un dataframe amb les metadades associades a les columnes del dataset anomenat col_data amb els identificadors de les mostres (SampleID):

```
col_data <- DataFrame(SampleID = colnames(assay_data))
```

Finalment, he creat l'objecte SummarizedExperiment amb les dades quantitatives (assays), les metadades dels metabolits (rowData) i les metadades de les mostres (colData):

```
se <- SummarizedExperiment( assays = list(counts = assay_data),
rowData = row_data, colData = col_data )
se
```

```
## class: SummarizedExperiment
## dim: 1438 12
## metadata(0):
## assays(1): counts
## rownames: NULL
## rowData names(6): SequenceModifications Accession ... CLASS PHOSPHO
## colnames(12): M1_1_MSS M1_2_MSS ... M64_1_PD M64_2_PD
## colData names(1): SampleID
```

```
save(se, file="summarized_experiment.Rda")
```

Les principals diferències entre la classe SummarizedExperiment i la classe ExpressionSet es troben en l'estructura de cada classe i el seu enfocament per a l'emmagatzematge i gestió de dades d'experimentació genòmica o proteòmica. Mentre que ExpressionSet es va dissenyar principalment per gestionar les dades de microarrays i s'ha utilitzat àmpliament en l'anàlisi de dades d'expressió gènica, SummarizedExperiment és més flexible i es va generar per adaptar-se a una varietat més àmplia de tipus de dades (e.g. NGS, proteòmica). Una diferència important entre les dues classes és que SummarizedExperiment permet l'emmagatzematge de diversos conjunts de dades quantificades o assays (e.g. condicions experimentals o mostres) dins d'un únic objecte, mentre que ExpressionSet està limitat a un sol conjunt d'assays per objecte. A més a més, SummarizedExperiment utilitza una estructura basada en rowData per a metadades associades amb les files (en aquest cas, metabòlits) i colData per a metadades relacionades amb les columnes (en aquest cas, les mostres), mentre que en un ExpressionSet, les metadades de les mostres s'emmagatzamarien de manera diferent i serien més rígides. Finalment, la classe SummarizedExperiment també ofereix una millor integració amb altres paquets i eines Bioconductor més recents, millorant la seva interoperabilitat per a l'anàlisi de dades òmiques complexes.

Anàlisi exploratori i estadístic del dataset:

Abans de començar a explorar el dataset, he carregat les llibreries SummarizedExperiment per treballar amb l'objecte se i ggplot2 per realitzar gràfics.

```
library(SummarizedExperiment)
library(ggplot2)
```

En aquest dataset es recullen les dades d'espectrometria de masses de 1438 fosfopèptides a 6 mostres de PDX (12 mostres amb els duplicats tècnics respectius). En primer lloc, per explorar les dimensions i contingut del dataset, faig servir l'objecte SummarizedExperiment que he generat a l'apartat anterior (Annex 1).Es confirma que tenim 1438 metabolits i 12 mostres a l'objecte SummarizedExperiment. La informació associada amb les files està recollida en 6 columnes (sequencemodifications, accession, description, score, class i phospho) i l'associada a les columnes només en una anomenada SampleID. A continuació, comprovo si hi ha valors nuls o amb variància zero per eliminar-los del conjunt:

```
sum(is.na(assay(se)))
```

```
## [1] 0
```

```
row_variances <- apply(assay(se), 1, var)
zero_var_rows <- sum(row_variances == 0)
print(zero_var_rows)
```

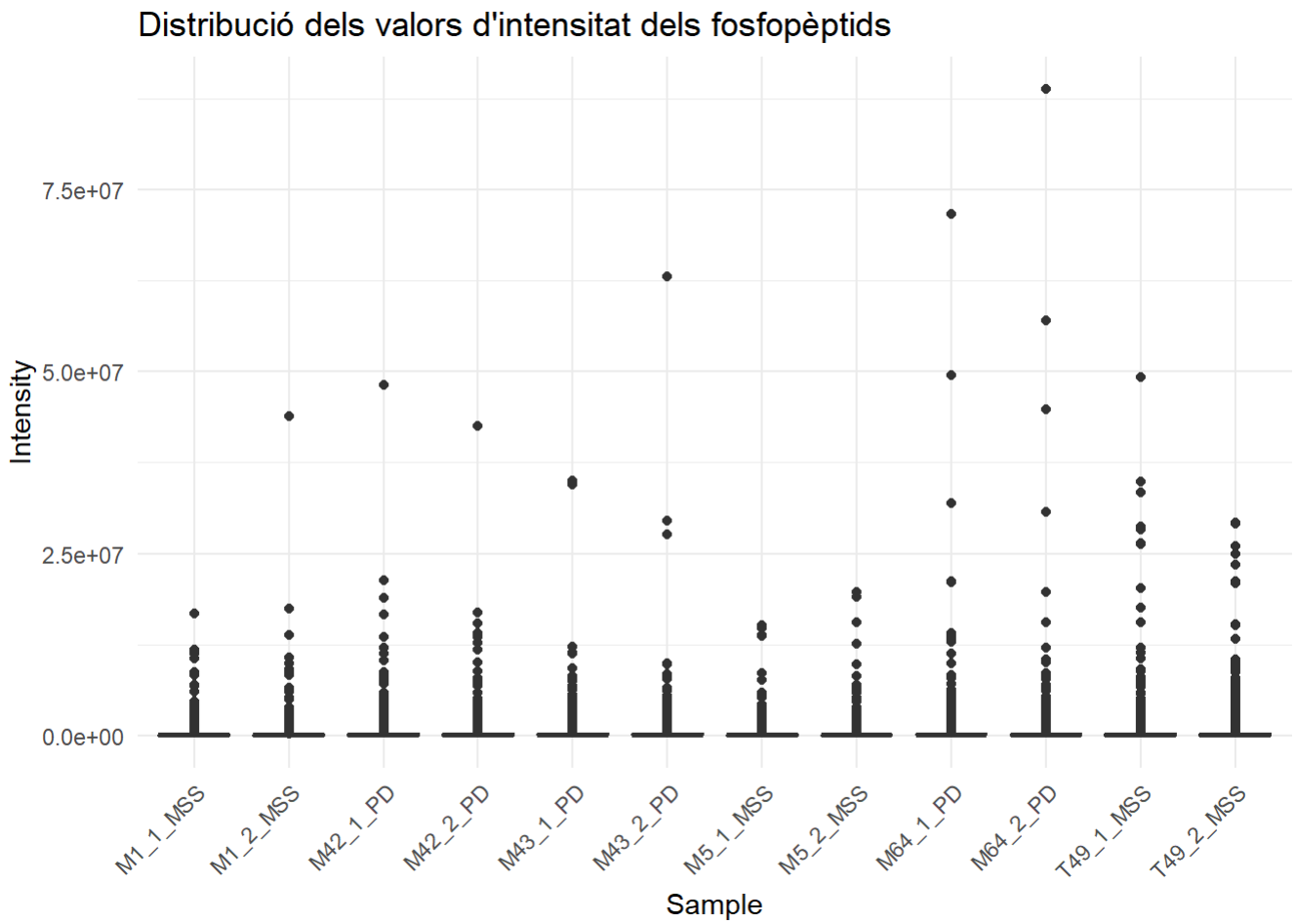
```
## [1] 2
```

```
filtered_se <- se[row_variances > 0, ]
```

No hi ha cap valor nul al dataset però trobem dos valors amb variança 0 i s'eliminen del se.

Tot seguit, per visualitzar de manera preliminar la distribució dels valors d'intensitat dels metabolits en cada mostra i identificar la presència de valors extrems i potencials diferències entre mostres he graficat les dades en format boxplot amb els valors d'intensitat dels metabolits a l'eix Y i l'ID de les mostres a l'eix X. Per tal de crear el gràfic, primerament, he creat un dataframe amb 2 columnes (valors d'intensitat i IDs de les mostres) i, a continuació, he generat el gràfic amb la llibreria ggplot2:

```
df <- data.frame(Intensity = as.vector(assay(filtered_se)),
                 Sample = rep(colnames(filtered_se), each = nrow(filtered_se)))

ggplot(df, aes(x = Sample, y = Intensity)) +
  geom_boxplot(fill = "lightblue") +
  theme_minimal() +
  labs(title = "Distribució dels valors d'intensitat dels fosfopèptids") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



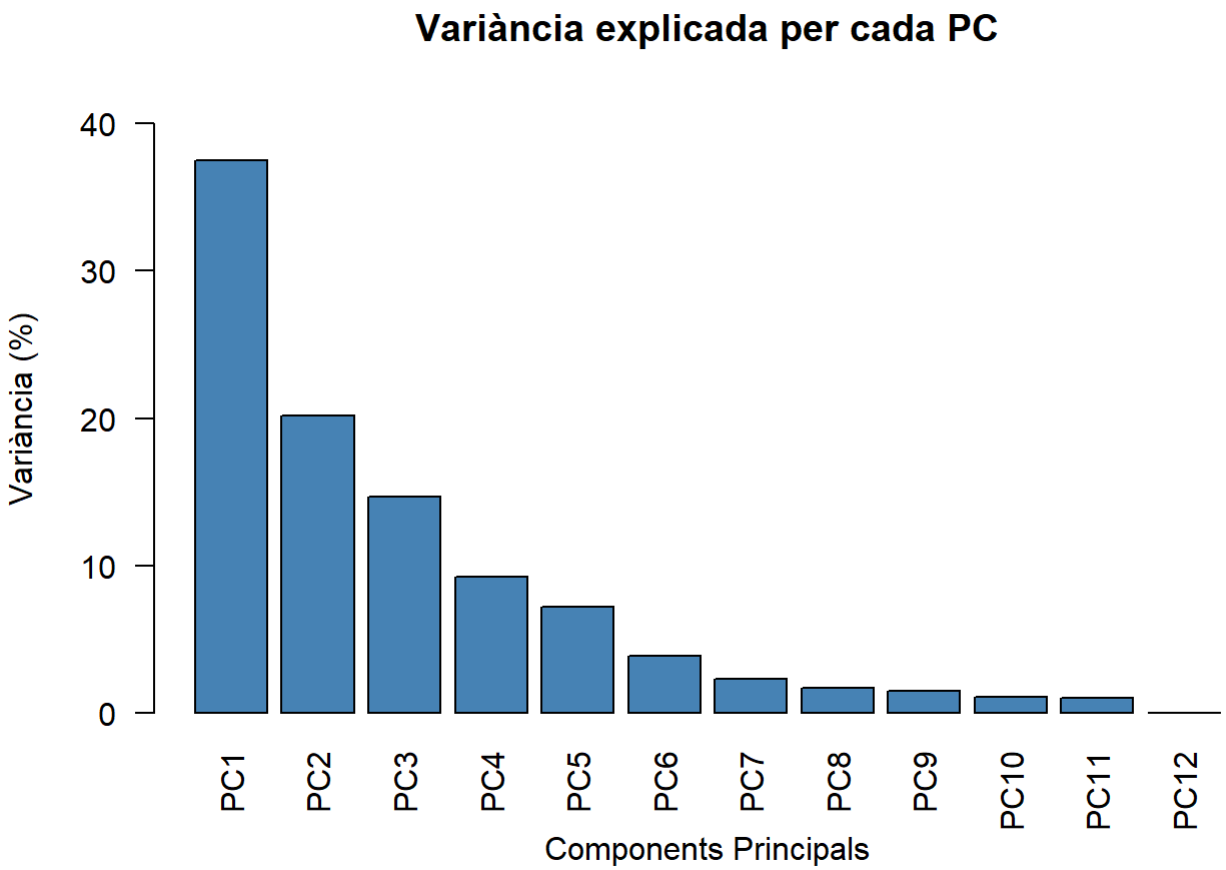
El gràfic generat però no permet distingir els fosfopèptids ja que se n'han analitzat 1438 per mostra ni veure clarament les diferències entre mostres.Com que es tracta d'un dataset molt gran i complex (1438 metabolits per 12 mostres), i el boxplot per si sol no m'ha permès identificar patrons clars, decideixo fer PCA per tal de reduir la dimensionalitat del dataset, esbrinar si existeixen diferents fenotips de tumor i quins metabolits poden ser determinants per la seva separació.

Realitzo el PCA sobre el se amb les dades filtrades i havent eliminat els valors amb variança 0. Per tal de veure el nombre de components identificats en l'anàlisi i el percentatge de variància associat a cadascun també calculo la variància en percentatge i la grafico en format barplot:

```
pca_res <- prcomp(t(assay(filtered_se)), scale. = TRUE)
explained_variance <- (pca_res$sdev^2) / sum(pca_res$sdev^2) * 100
explained_variance_df <- data.frame(
     PC = paste0("PC", 1:length(explained_variance)),
     Variance = explained_variance)
print(explained_variance_df)
```

```
##       PC     Variance
## 1    PC1 3.750520e+01
## 2    PC2 2.015073e+01
## 3    PC3 1.463183e+01
## 4    PC4 9.242100e+00
## 5    PC5 7.200322e+00
## 6    PC6 3.819626e+00
## 7    PC7 2.259561e+00
## 8    PC8 1.696193e+00
## 9    PC9 1.456803e+00
## 10 PC10 1.072883e+00
## 11 PC11 9.647575e-01
## 12 PC12 1.295372e-29
```

```
pc_labels <- paste0("PC", 1:length(explained_variance))
barplot(explained_variance, names.arg = pc_labels, main = "Variància explicada per cada PC",
        xlab = "Components Principals", ylab = "Variància (%)", col = "steelblue",
        ylim = c(0, max(explained_variance) * 1.1), las = 2)
```



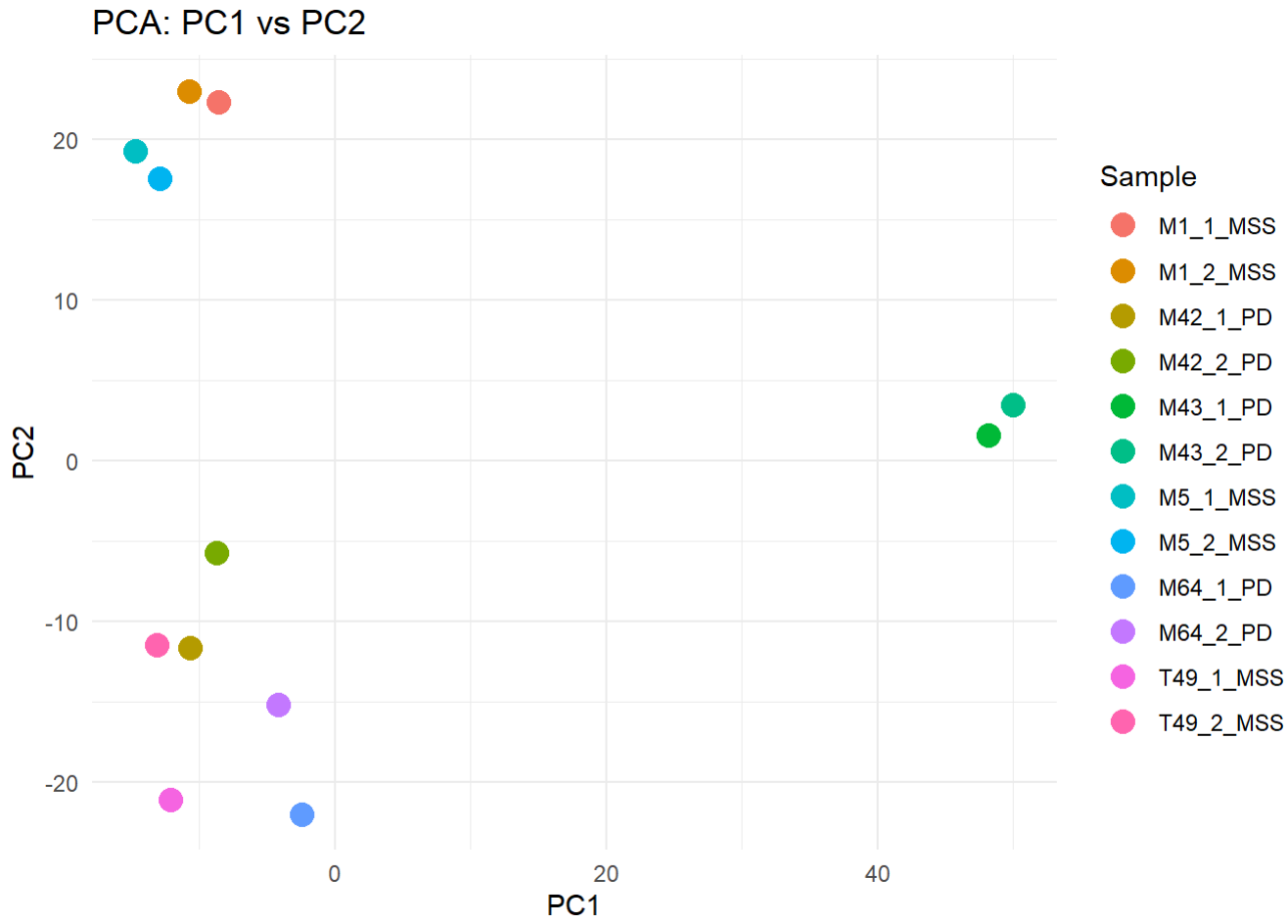
Segons aquests resultats, PC1, PC2, PC3 i PC4 concentren la major part de la variança amb 37.5%, 20.1%, 14.6% i 9.2%, respectivament.Com que PC4 només explica una petita fracció de la variància (<10%), és probable que no tingui un impacte fort en la separació de les mostres, així que em centro en els tres primers components identificats.En primer lloc, vull comprovar si els tres components estan correlacionats o si pel contrari capturen informació independent:

```
correlation_matrix <- cor(pca_res$x[, 1:3])
print(correlation_matrix)
```

```
##               PC1           PC2           PC3
## PC1  1.000000e+00 -1.747401e-16 -5.587156e-16
## PC2 -1.747401e-16  1.000000e+00 -2.180859e-16
## PC3 -5.587156e-16 -2.180859e-16  1.000000e+00
```

Els valors dins de la matriu de correlació fora de la diagonal són pràcticament 0, la qual cosa indica que PC1, PC2 i PC3 no estan correlacionats entre si. Un cop determinat això, comprovo com separen les mostres els components PC1 i PC2 determinats al PCA:

```
pca_df <- data.frame(PC1 = pca_res$x[,1], PC2 = pca_res$x[,2], Sample = colData(filtered_se)
$SampleID)

ggplot(pca_df, aes(x = PC1, y = PC2, color = Sample)) +
  geom_point(size = 4) +
  theme_minimal() +
  labs(title = "PCA: PC1 vs PC2")
```



S'observa una separació de mostres al llarg del component 2 (M1 i M5 a la part superior, i T49, M64 i M42 a la part inferior). La mostra M43 es separa dels dos grups situant-se al centre de l'eix i desplaçada a la dreta del l'eix X mentre que la resta de mostres es situen a l'inici de l'eix. Els replicats de cada mostra estan relativament junts dins de l'espai del PCA, i per tant és poc probable que estiguin influenciant la separació per components. Tot seguit, comprovo si l'addició de PC3 millora la separació de les mostres:

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

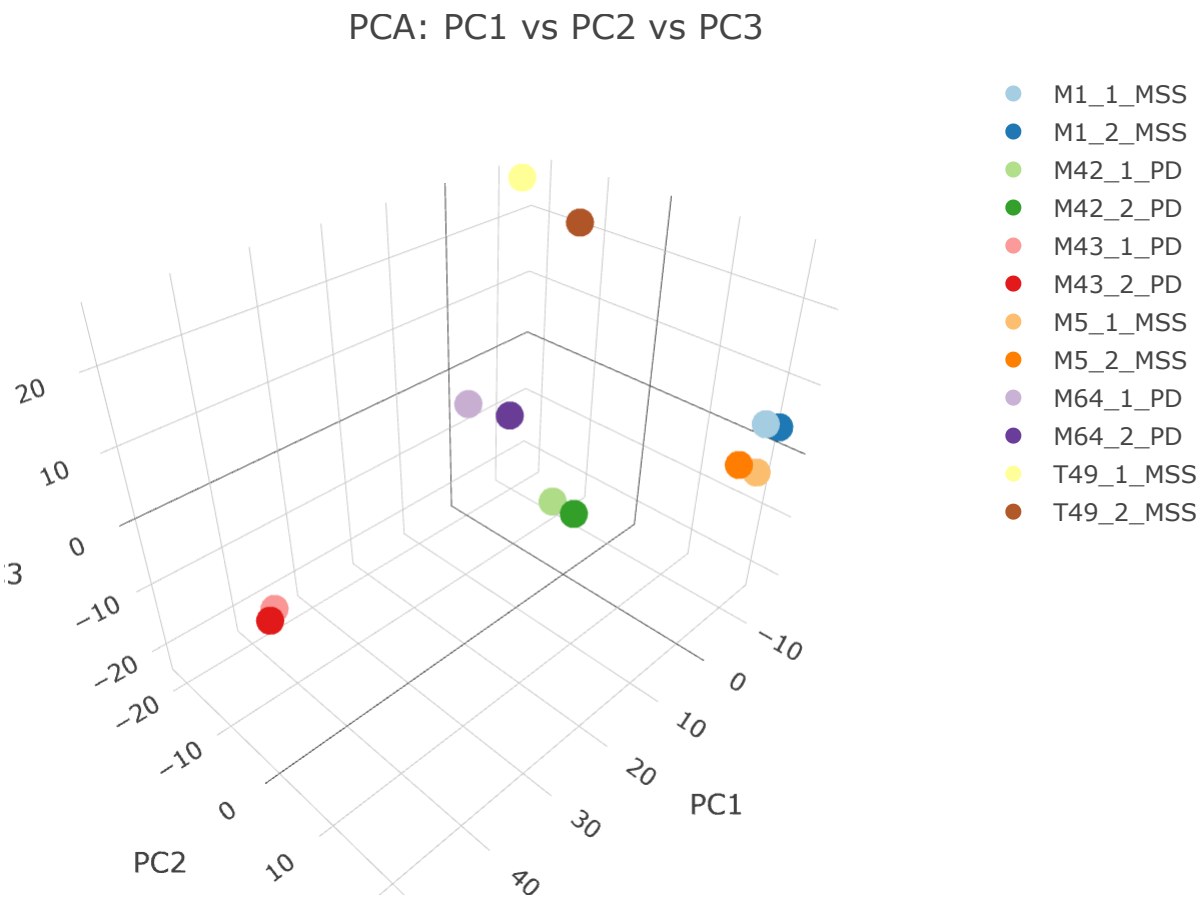
```
## The following object is masked from 'package:IRanges':
##
##     slice
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

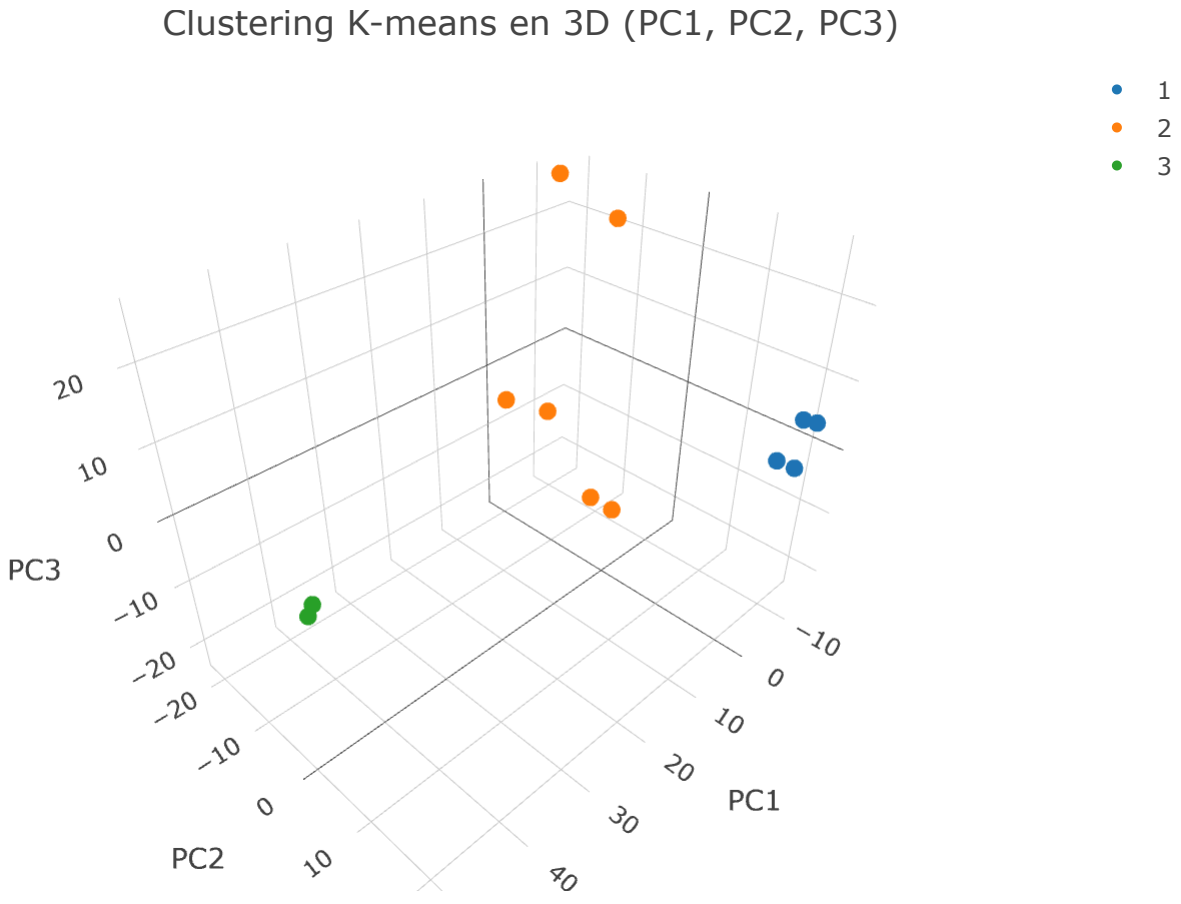
```
df_pca <- data.frame(pca_res$x[, 1:3], SampleID = col_data$SampleID)

plot_ly(df_pca, x = ~PC1, y = ~PC2, z = ~PC3, color = ~SampleID, colors = "Paired") %>%
  add_markers() %>%
  layout(title = "PCA: PC1 vs PC2 vs PC3")
```



En aquest cas també observo 3 subgrups de mostres de manera similar al gràfic anterior i visualment trobo difícil decidir si el component PC3 millora la separació dels subgrups de manera significativa. Per quantificar la separació entre mostres determino la distància euclidiana per PC1 vs PC2 i PC1 vs PC2 vs PC3 (Annex 2). El resultat demostra que tant la distància mitjana com la màxima augmenten en 3D i, per tant, PC3 aporta una certa variabilitat addicional que ajuda a separar millor les mostres en l'espai PCA.

A través de la reducció de la dimensionalitat associada a PCA he observat la separació de les mostres tumorals en 3 subgrups: M1, i M5; T49, M64 i M42 i, finalment, M43. Per confirmar l'existència d'aquests tres subgrups de mostres, he dut a terme un anàlisis de clustering k-means assumint l'existència de 3 clusters de mostres:

```
pca_scores <- as.data.frame(pca_res$x)
pca_scores$Sample <- rownames(pca_scores)
set.seed(123)
num_clusters <- 3
kmeans_result <- kmeans(pca_scores[, 1:3], centers = num_clusters)
clustered_data <- as.data.frame(pca_scores)
clustered_data$Cluster <- as.factor(kmeans_result$cluster)
fig <- plot_ly(clustered_data, x = ~PC1, y = ~PC2, z = ~PC3,
               color = ~Cluster, colors = c('#1f77b4', '#ff7f0e', '#2ca02c'),
               type = 'scatter3d', mode = 'markers', marker = list(size = 5)) %>%
               layout(title = "Clustering K-means en 3D (PC1, PC2, PC3)",
               scene = list(xaxis = list(title = 'PC1'),
                       yaxis = list(title = 'PC2'),
                        zaxis = list(title = 'PC3')))
fig
```



L'anàlisi clustering de k-means confirma tres subgrups de mostres tumorals (M1, i M5; T49, M64 i M42 i, M43). Finalment, per identificar quins fosfopèptids tenen més influència sobre cadascun d'aquests tres components (PC1, PC2 i PC3) i, per tant, sobre la separació de les mostres, he determinat els loadings de cada variable original sobre aquests components i l'he relacionat amb el nom del fosfopèptid (SequenceModifications) (Annex 3). A través d'aquest anàlisi he identificat els 10 fosfopèptids amb més pes sobre cadascun d'aquests tres components. Cap dels fosfopèptids identificats està present en més d'un component i, per tant, es pot concloure que el seu pes sobre el percentage de la variància de cada component no està correlacionat.

Discussió

Els resultats de PCA han demostrat que els components PC1, PC2 i PC3 expliquen el 37.5%, 20.1% i 14.6% de la variància, respectivament. La visualització de les mostres a través de PCA nostra una separació clara de mostres, amb tres agrupaments que coincidien amb fenotips tumorals específics: un grup per a les mostres M1 i M5, un altre per a les mostres T49, M64 i M42, i un grup intermedi que inclou la mostra M43. Aquests resultats, reforçats pels de clustering amb k-means, suggereixen que les modificacions posttraduccionals com la fosforilació poden jugar un paper fonamental en la diferenciació entre fenotips tumorals. Tot i que l'anàlisi PCA ha mostrat una bona separació entre les mostres, cal destacar que l'addició de PC3 aporta informació addicional que ajuda a millorar la separació, encara que no es pot determinar de manera concloent si aquesta tercera component és crítica per a la classificació. Posteriorment, s'ha assignat un cluster a cada subgrup de mostres identificat per PCA i s'ha partit d'un nombre de clusters = 3. No obstant, per tal de determinar el nombre òptim de clusters per l'anàlisi es podria haver calculat la variabilitat dins de cada cluster (wss) en funció del nombre de clusters. Això ens hagués servit per optimitzar els paràmetres del PCA, encara que 3 clusters han demostrat ser visualment suficients per separar els 3 subgrups de mostres. Encara que l'anàlisi per PCA ha identificat 3 subgrups de fenotip tumoral a nivell fosfoproteòmic cal remarcar que el nombre de mostres analitzat per fosfoproteòmica és molt baix (n=6 amb 2 replicats tècnics). Un dels subgrups identificats només està constituit per una mostra tumoral. Caldria incrementar la n de mostres per confirmar que si es tracta d'un fenotip establert o si es tracta de valors inusuals o afectats per errors en l'anàlisi per MS.
Els fosfopèptids més influents identificats en els tres components principals poden proporcionar indicis sobre quines modificacions específiques tenen més impacte en les diferències entre els subgrups tumorals. Aquestes troballes obren la porta a noves línies d'investigació sobre els mecanismes moleculars darrere de les diferències fenotípiques observades.És important emfatitzar que cap dels fosfopèptids associats amb els components del PCA està present a la llista associada amb els de cap altre.

Conclusions

En aquest estudi, l'ús de tècniques d'anàlisi de reducció de dimensionalitat com el PCA i el clustering k-means ha permès identificar agrupaments clars dins de les mostres tumorals en funció dels seus nivells de fosfopèptids, suggerint l'existència de tres fenotips diferenciats. Els fosfopèptids més influents identificats podrien ser candidats per a futurs estudis sobre biomarcadors relacionats amb la fosforilació. Aquest enfocament bioinformàtic proporciona una eina prometedora per a l'anàlisi de dades òmiques complexes amb implicacions potencials en la identificació de biomarcadors associats amb el càncer.

Referències

Enllaç al repositori Github: https://github.com/jbagunatorres/Phosphoproteomics-analysis.git (https://github.com/jbagunatorres/Phosphoproteomics-analysis.git)

Annexos

Annex 1

```
dim(se)
```

```
## [1] 1438   12
```

```
colData(se)
```

```
## DataFrame with 12 rows and 1 column
##              SampleID
##           <character>
## M1_1_MSS     M1_1_MSS
## M1_2_MSS     M1_2_MSS
## M5_1_MSS     M5_1_MSS
## M5_2_MSS     M5_2_MSS
## T49_1_MSS   T49_1_MSS
## ...               ...
## M42_2_PD     M42_2_PD
## M43_1_PD     M43_1_PD
## M43_2_PD     M43_2_PD
## M64_1_PD     M64_1_PD
## M64_2_PD     M64_2_PD
```

```
rowData(se)
```

```
## DataFrame with 1438 rows and 6 columns
##       SequenceModifications   Accession            Description     Score
##                   <character> <character>            <character> <numeric>
## 1    LYPELSQYMGLSLNEEEIR[..      O00560 Syntenin-1 OS=Homo s..     48.07
## 2    VDKVIQAQTAFSANPANPAI..      O00560 Syntenin-1 OS=Homo s..     67.05
## 3    VIQAQTAFSANPANPAILSE..      O00560 Syntenin-1 OS=Homo s..     77.71
## 4    HADAEMTGYVVTR[6] Oxi..      O15264 Mitogen-activated pr..     44.87
## 5    HADAEMTGYVVTR[9] Pho..      O15264 Mitogen-activated pr..     67.42
## ...                     ...         ...                     ...       ...
## 1434 YLLSQSSPAPLTAAEEELR[..      Q12792 Twinfilin-1 OS=Homo ..     56.19
## 1435   YLSFTPPEK[3] Phospho      Q13177 Serine/threonine-pro..     39.14
## 1436 YNLDASEEEDSNK[6] Pho..      O95218 Zinc finger Ran-bind..     80.66
## 1437 YQDEVFGGFVTEPQEESEEE..      Q13283 Ras GTPase-activatin..     40.01
## 1438 YSPSQNSPIHHIPSRR[1] ..      Q9NYF8 Bcl-2-associated tra..     36.71
##            CLASS     PHOSPHO
##      <character> <character>
## 1              H           Y
## 2              H           Y
## 3              H           Y
## 4              H           Y
## 5              H           Y
## ...          ...         ...
## 1434           C         S/T
## 1435           C         S/T
## 1436           C         S/T
## 1437           C         S/T
## 1438           C         S/T
```

Annex 2

```
dist_2d <- dist(df_pca[, c("PC1", "PC2")])
dist_3d <- dist(df_pca[, c("PC1", "PC2", "PC3")])
summary(dist_2d)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.316  10.032  34.369  34.353  57.360  66.755
```

```
summary(dist_3d)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.569  31.040  42.067  41.304  58.597  70.860
```

Annex 3

```
rownames(se) <- rowData(se)$SequenceModifications
rownames(filtered_se) <- rownames(se)[row_variances > 0]
row_data_filtered <- rowData(se)[rownames(filtered_se), ]
loadings_df <- as.data.frame(pca_res$rotation)
loadings_df$Metabolite <- rownames(filtered_se)
loadings_df <- cbind(loadings_df, row_data_filtered)
top_pc1 <- loadings_df[order(abs(loadings_df$PC1), decreasing = TRUE), ]
top_metabolites_pc1 <- top_pc1$SequenceModifications
print(head(top_metabolites_pc1, 10))
```

```
##  [1] "EEKDKDDQEWESPSPPKPTVFISGVIAR[14] Phospho"
##  [2] "QKSEEPSVSIPFLQTALLR[9] Phospho"
##  [3] "IGSDPLAYEPK[3] Phospho"
##  [4] "VDINTPDVDVHGPDWHLK[5] Phospho"
##  [5] "DKDDQEWESPSPPKPTVFISGVIAR[11] Phospho"
##  [6] "TASELLLDR[3] Phospho"
##  [7] "RPTPNDDTLDEGVGLVHSNIATEHIPSPAK[27] Phospho"
##  [8] "STENVNKLPELSSFGKPSSSVQGTGLIR[2] Phospho"
##  [9] "WLDESDAEMELR[5] Phospho"
## [10] "TSPKPAVVETVTTAKPQQIQALMDEVTK[2] Phospho"
```

```
top_pc2 <- loadings_df[order(abs(loadings_df$PC2), decreasing = TRUE), ]
top_metabolites_pc2 <- top_pc2$SequenceModifications
print(head(top_metabolites_pc2, 10))
```

```
##  [1] "ADEASELACPTPK[9] Carbamidomethyl|[11] Phospho"
##  [2] "AQLEPVASPAK[8] Phospho"
##  [3] "RPEGPGAQAPSSPR[12] Phospho"
##  [4] "DELHIVEAEAMNYEGSPIK[11] Oxidation|[16] Phospho"
##  [5] "GEPNVSYICSR[7] Phospho|[9] Carbamidomethyl"
##  [6] "QSPEDVYFSK[7] Phospho"
##  [7] "NKSNEDQSMGNWQIK[3] Phospho|[9] Oxidation"
##  [8] "HVSPVTPPR[3] Phospho"
##  [9] "RPSQEQSASASSGQPQAPLNR[7] Phospho"
## [10] "ESEDKPEIEDVGSDEEEEK[13] Phospho"
```

```
top_pc3 <- loadings_df[order(abs(loadings_df$PC3), decreasing = TRUE), ]
top_metabolites_pc3 <- top_pc3$SequenceModifications
print(head(top_metabolites_pc3, 10))
```

```
##  [1] "KDEETEESEYDSEHENSEPVTNIR[12] Phospho"
##  [2] "EKLQEEGGGSDEEETGSPSEDGMQSAR[10] Phospho"
##  [3] "EYIPGQPPLSQSSDSSPTR[10] Phospho"
##  [4] "RLEISPDSSPER[5] Phospho"
##  [5] "SGLTVPTSPK[7] Phospho"
##  [6] "AHLTVGQAAAGGSGNLLTER[13] Phospho"
##  [7] "APLKPYPVSPSDK[9] Phospho"
##  [8] "FTDKDQQPSGSEGEDDDAEAALKK[9] Phospho"
##  [9] "SQSLPNSLDYTQTSDPGR[3] Phospho"
## [10] "NVPQEESLEDSDVDADFK[11] Phospho"
```