PRINCIPE FELIPE
CENTRO DE INVESTIGACION

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Grado en Biotecnología

ETSIAMN- Universitat Politècnica de València

Trabajo Fin de Grado - Curso 2016/2017

ESTUDIO DEL EFECTO DE LA EXPOSICIÓN A PESTICIDAS EN EL NEURODESARROLLO MEDIANTE LA INTEGRACIÓN DE DATOS ÓMICOS

Valencia, Junio 2017

ALUMNO

Víctor Sánchez Gaya

TUTORA

Sonia Tarazona Campos

COTUTORES EXTERNOS

Ana Conesa Cegarra

Vicente Felipo Orts

Título: Estudio del efecto de la exposición a pesticidas en el neurodesarrollo mediante la integración de datos ómicos

Autor: Víctor Sánchez Gaya

Tutora: Sonia Tarazona Campos

Cotutores externos: Ana Conesa Cegarra y Vicente Felipo Orts

Localidad y fecha: Valencia, Junio de 2017

Resumen

El enfoque multiómico está ganando terreno en el campo de la biología de sistemas, ya que permite estudiar aspectos complementarios del funcionamiento de un sistema biológico. Sin embargo, la integración de datos ómicos en un modelo estadístico que ayude a entender los mecanismos de regulación en la célula es todavía un reto importante en bioinformática.

En este proyecto, en el que colaboran los laboratorios de Genómica de la Expresión Génica y de Neurobiología del Centro de Investigación Príncipe Felipe, se ha abordado el análisis de datos multiómicos (proteómica, metabolómica) procedentes del proyecto europeo DENAMIC, en el que se investigaron los efectos neurotóxicos de distintos pesticidas de uso común en crías de ratas expuestas durante el embarazo y lactancia, integrándolos con los resultados de diversos tests que medían capacidades cognitivas y motoras.

El objetivo de este trabajo es procesar los datos y realizar los análisis estadísticos apropiados que permitan integrar toda la información para entender los mecanismos celulares subyacentes al proceso biológico estudiado, así como interpretar los resultados obtenidos desde el punto de vista biológico para poder obtener finalmente un conjunto de biomarcadores sensibles y representativos de alteraciones cognitivas o motoras.

Palabras clave- Análisis de integración multiómico, biomarcadores, proteómica, metabolómica, bioinformática, neurobiología, alteraciones cognitivas, alteraciones motoras, exposición a pesticidas.

Summary

The multi-omic approach is gaining ground in the field of systems biology, since it allows to study complementary aspects of how a biological system works. However, the integration of omic data into a statistical model that helps to understand the mechanisms of regulation in the cell is still a major challenge in bioinformatics.

This project, which is a collaboration between the Genomics of Gene Expression Laboratory and the Neurobiology Laboratory at the Príncipe Felipe Research Center, tackles the analysis of multi-omic data (proteomics and metabolomics) from the European project DENAMIC, in which the neurotoxic effects of several commonly used pesticides on the offspring of rats exposed during pregnancy and lactation were studied by means of integrating these multi-omic data with the results of various tests that measured cognitive and motor skills.

The goal of this work is to process the data and to perform the appropriate statistical analyses that allow to integrate all the information to understand the cellular mechanisms underlying the biological process studied, as well as to interpret the results obtained from the biological point of view in order to finally obtain a set of sensitive and representative biomarkers for cognitive or motor alterations.

Keywords- Multi-omic integrative analysis, biomarkers, proteomics, metabolomics, bioinformatics, neurobiology, cognitive alterations, motor alterations, exposure to pesticides.

Resum

L'aproximació multiòmica està guanyant terreny en el camp de la biologia de sistemes, ja que permet estudiar aspectes complementaris del funcionament d'un sistema biològic. No obstant això, la integració de dades òmiques en un model estadístic que ajude a entendre els mecanismes de regulació en la cèl·lula és encara un repte important en bioinformàtica.

En aquest projecte, en el qual col·laboren els laboratoris de Genòmica de l'Expressió Gènica i de Neurobiologia del Centre d'Investigació Príncep Felip, s'ha abordat l'anàlisi de dades multiòmiques (proteòmica, metabolòmica) procedents del projecte europeu DENAMIC, en el qual es van investigar els efectes neurotòxics de diferents pesticides d'ús comú en cries de rates exposades durant l'embaràs i la lactància, integrant-los amb els resultats de diversos tests que mesuraven capacitats cognitives i motores.

L'objectiu d'aquest treball és processar les dades i realitzar les anàlisis estadístiques apropiades que permeten integrar tota la informació per entendre els mecanismes cel·lulars subjacents al procés biològic estudiat, així com interpretar els resultats obtinguts des del punt de vista biològic per poder obtenir finalment un conjunt de biomarcadors sensibles i representatius d'alteracions cognitives o motores.

Paraules clau- Anàlisi d'integració multiòmica, biomarcadors, proteòmica, metabolòmica, bioinformàtica, neurobiologia, alteracions cognitives, alteracions motores, exposició a pesticides.

Agradecimientos

Ha sido todo un placer y un privilegio poder formar parte del grupo de investigación de Genómica de la Expresión Génica liderado por Ana Conesa. Privilegio en lo profesional, y placer en lo personal. Por lo que me gustaría agradecer a Ana, como figura presidencial, la oportunidad de haberme dejado formar parte de su equipo.

Me gustaría agradecer también especialmente el trabajo, atención y paciencia de mi entrenadora, Sonia. Me quito el sombrero, gracias.

Por último, a mi equipo técnico, a la gente que no aparece en las portadas pero sin los cuáles tampoco hubiera podido conseguir nada. Por un lado a Manu y a Carlos del departamento de soporte técnico, estratégico y de análisis de resultados a pie de pista, aunque no sé si debería haberles mencionado, ya que se comen todas mis rosquilletas. A Mireia, la expatriada que me da consejos desde la lejanía. A mi nutricionista personal durante estos últimos 4 años, la yaya Pilar, sigo sin creerme que no haya aumentado mi peso más de 200 kg. A mi principal grupo inversor, también llamados padres, que siempre están ahí para apoyarme y animarme a seguir a delante, y cuando digo siempre, es siempre. A aquellas personas que aunque no hayan podido llegar a verme terminar la carrera han contribuido enormemente para ello, os echo de menos. Y como no, a mi compitrueno María, probablemente sin su ayuda seguiría estancado en 2º de carrera.

De corazón, gracias.

Índice general

Índice de Figuras

Índice de tablas

Abreviaturas

C1	Primera componente
C2	Segunda componente
CAR	Carbaril
CB	Cerebelo
CIPF	Centro de Investigación Príncipe Felipe
CHLOR	Clorpirifos
CHLOR01	Clorpirifos 0.1 mg/kg/día
CHLOR03	Clorpirifos 0.3 mg/kg/día
CHLOR1	Clorpirifos 1 mg/kg/día
CV	Coeficiente de variación
CYP	Cypermetrín
D7G	Día 7 de la gestación
D	Día
DENAMIC	*Developmental Neurotoxicity Assesment of Mixtures In Children* (Evaluación neurotóxica de mezclas en el desarrollo neurológico de niños)
D.E.	Diferencialmente expresados o expresadas
END	Endosulfán
GC-HRTOF-MS	*Gas Chromatography High Resolution Time of Flight Mass Spectrometry* (Cromatografía gaseosa - tiempo de vuelo de alta resolución - espectrometría de masas)
HA	Hiperamonemia
HP	Hipocampo
LC-HRTOF-MS	*Liquid Chromatography High Resolution Time of Flight Mass Spectrometry* (Cromatografía líquida - tiempo de vuelo de alta resolución - espectrometría de masas
MSEP	*Mean Square Error of Prediction* (Error de predicción cuadrático medio)
MWM	Morris Water Maze
OBS	Observaciones
PN	Postnatal
PC	Componente principal
PCA	Análisis de componentes principales
PLS	*Partial Least Squares* (Regresión por mínimos cuadrados parciales)
RM	Radial Maze
Rot	Rotarod
RMRE	Radial Maze errores de referencia
RMWE	Radial Maze errores de trabajo
T.S.	Total por sexo
TFG	Trabajo de Fin de Grado
TNF	*Tumor Necrosis Factor* (Factor de necrosis tumoral)
VF	Valor Faltante
VFs	Valores Faltantes
VIP	*Variable Importance in Projection* **(**Importancia de las variables en la proyección**)**

1-Introducción

1.1 Disciplinas ómicas y su desarrollo

La llegada de las nuevas técnicas de secuenciación ha conllevado la generación de datos de índole masiva en el campo de la biología. La gran cantidad de información aportada por estas nuevas metodologías ha permitido un importante desarrollo de las diferentes ómicas, estableciéndose la proteómica, la metabolómica la genómica y la transcriptómica como las más destacadas. Como consecuencia del mayúsculo tamaño de los datos se han tenido que aplicar y desarrollar diferentes herramientas informáticas y estadísticas para facilitar su preparado, procesado, y comprensión que han llevado a la bioinformática a consolidarse como uno de los campos con mayor futuro en la actualidad, en el ámbito de la biología. En consonancia con este avance se están implementando nuevas estrategias y metodologías para integrar la información aportada por las diferentes ómicas con tal de llevar a cabo estudios de tipo multiómico.

Atendiendo a las diferentes ómicas, en este proyecto se van a analizar datos proteómicos y metabolómicos.

1.2 Proteómica y Metabolómica

Se conoce como proteómica el estudio y caracterización de todo el conjunto de proteínas que se expresan en un genoma. El cúmulo de todas las proteínas presentes en una célula en un momento determinado recibe el nombre de proteoma. Uno de los principales objetivos de la proteómica es la identificación, caracterización y clasificación de las proteínas atendiendo a sus funciones y a las interacciones que se establecen entre ellas [1].

Por otro lado, la metabolómica hace referencia al estudio de los diversos sustratos y productos del metabolismo, los metabolitos [2]. El conjunto de todos los metabolitos que se pueden encontrar en una muestra en un instante concreto se conoce como metaboloma, de forma análoga al proteoma.

Mientras que el genoma de todas las células de un organismo es siempre el mismo, tanto el proteoma como el metaboloma varían en función del tipo celular así como de los factores externos o internos que actúan sobre las mismas células, pudiendo presentar por ello una misma célula un gran número de proteomas y metabolomas frente a un único genoma. Son por ello el proteoma y el metaboloma de gran interés para revelar el estado actual de la muestra estudiada y los cambios que en ella se han producido frente a diferentes condiciones suponiendo por ello una gran fuente de conocimiento para muchos análisis.

1.3 Proyecto Denamic

1.3.1 Descripción global del proyecto

Este estudio forma parte del proyecto europeo DENAMIC (*Developmental Neurotoxicity Assesment of Mixtures In Children*) en el que se analizaron los efectos de la exposición a diferentes pesticidas durante la gestación y la lactancia sobre la función cognitiva y motora de los niños. Durante la lactancia y edades más tempranas se lleva a cabo la parte principal del desarrollo cerebral. Mientras transcurre esta etapa el cerebro es muy sensible a los efectos nocivos que puedan tener determinados contaminantes que les pueden ocasionar efectos de larga duración e incluso permanentes [3]. Por ello los resultados obtenidos en DENAMIC pueden aportar información a la Unión Europea y a la Organización Mundial de la Salud para gestionar los riesgos de la exposición a contaminantes químicos, para de esta forma poder ayudar a la legislación sobre potenciales neurotóxicos y a determinar los niveles tolerables, en el caso de que los haya, de estas sustancias.

El trabajo desarrollado a lo largo de estas páginas se corresponde con un estudio de índole multiómica sobre datos obtenidos en un modelo animal y, en él, se analiza el efecto de la exposición de pesticidas en el desarrollo neurológico mediante la integración de diversas ómicas.

DENAMIC ha sido financiado por la Comisión Europea dentro del FP7 Medio Ambiente y posee un notorio carácter internacional, participando 14 socios de 10 países. El laboratorio de Neurobiología del Centro de Investigación Príncipe Felipe (CIPF), liderado por el Dr. Vicente Felipo, es uno de los socios de este proyecto.

1.3.2 Diseño experimental

El modelo animal empleado es *Rattus norvegicus.* Los posibles efectos de los pesticidas se estudiaron sobre la descendencia de ratas embarazadas las cuales fueron expuestas a pesticidas durante el embarazo y el período de lactancia. Todos los experimentos descritos a continuación fueron realizados por el laboratorio de Neurobiología del CIPF.

El pesticida se administraba diariamente inmerso en una gelatina dulce (MediGel Sucralose de ClearH20) que ingerían de forma sencilla las ratas a partir del día 7 de gestación (D7G) (día en que se produce la implantación del embrión) hasta el día 21 postnatal (D21PN), por lo que seguían transmitiendo el pesticida a las crías mediante la leche. A partir del día 21 postnatal las crías no volvieron a ser expuestas a los pesticidas. En la gelatina de las ratas control (VH) el pesticida fue sustituido por aceite de maíz.

Para tratar de apreciar los cambios producidos por la acción de los pesticidas las ratas fueron sometidas a diferentes tests para valorar sus capacidades cognitivas y motoras cuando eran adultos jóvenes (2-3 meses), posteriormente se sacrificaron para para la obtención de muestras cerebrales de diversos tejidos: cerebelo (CB) e hipocampo (HP) , que fueron utilizadas para la generación de distintos datos ómicos (proteómica, metabolómica y transcriptómica) algunos de los cuales han sido analizados en este trabajo. La progresión temporal del estudio aparece representada en la Figura 1.

Se consideró también el efecto del sexo de las ratas a la hora de realizar los análisis y la interpretación de los resultados, debido a las grandes diferencias existentes a nivel bioquímico y de comportamiento entre machos y hembras y que pueden resultar en respuestas muy diferentes para un mismo tratamiento.

Debido a la imposibilidad de mantener vivas simultáneamente todas las ratas que se utilizaron en este estudio, por limitaciones de espacio, el trabajo se desarrolló por conjuntos de las mismas, aplicando sobre las ratas de cada conjunto (*set*) algunos de los pesticidas evaluados, según lo expuesto anteriormente en este apartado.

Por último, con los datos de las diferentes ómicas y los tests recopilados se desarrolló, en el laboratorio de Genómica de la Expresión Génica del CIPF, un análisis integrativo de esta información para la obtención de resultados.

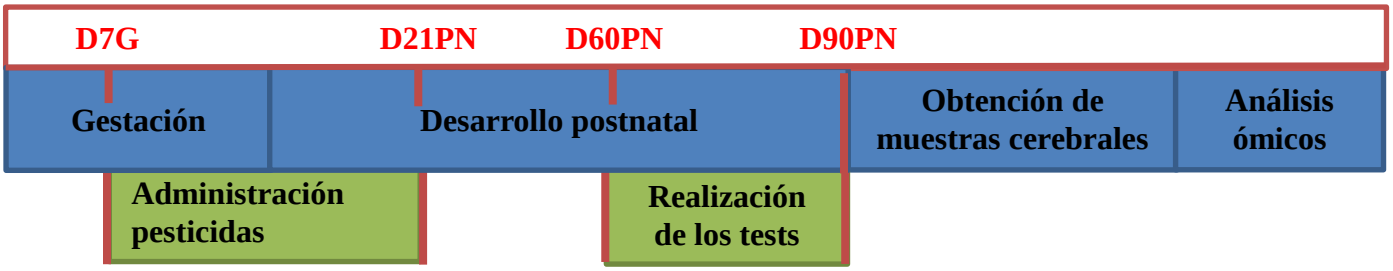


Figura 1. Progresión temporal del estudio desde el inicio de la gestación hasta la aplicación de los análisis ómicos. D (día), PN (postnatal), G (gestación)

1.3.3 Datos empleados en este trabajo: ómicas, pesticidas y tests.

La información de partida utilizada en este proyecto de fin de grado corresponde a dos de los *sets* de ratas que se analizaron en DENAMIC: el Set01 y el Set03.

Atendiendo a las ómicas, el Set01 constaba de datos de proteómica y metabolómica mientras que el Set03 incluía únicamente proteómica. Además, los pesticidas empleados en cada test también variaban, encontrando en el Set03: Clorpirifos (CHLOR) en concentraciones de 0.1, 0.3, y 1 mg/kg/día, (CHLOR01, CHLOR03 y CHLOR1), Carbaril (CAR) a 15 mg/kg/día, Cypermetrín (CYP) a 1.5 mg/kg/día y en el Set01: Endosulfán (END) a 0.5 mg/kg/día y Cypermetrín a 1.5 mg/kg/día junto con los respectivos individuos control en cada set. Las concentraciones utilizadas fueron determinadas mediante la bibliografía por ser las menores con las que se habían detectado efectos

Con respecto a los tests, se utilizaron los mismos en ambos sets: Radial Maze (RM) y Morris Water Maze (MWM) para medir la capacidad cognitiva y Rotarod (Rot) y Beam Walking (BW) para la capacidad motora.

El test de Radial Maze se realiza en un laberinto de plexiglás de ocho brazos, en 4 de los cuales se sitúa comida. Para cada rata la comida se sitúa siempre en los mismos 4 brazos, tras unas sesiones de entrenamiento y familiarización de las ratas con el laberinto, se las introduce en el mismo con la finalidad de anotar cuánto tiempo tarda en obtener la comida de los 4 brazos. Si la rata tarda más de 3 minutos en obtener la comida de los cuatro brazos se la saca del recinto. Atendiendo al transcurso de la prueba se miden dos parámetros: el de errores de referencia “reference errors” (RMRE), que es el número de veces que la rata visita un brazo sin comida, y el de errores de trabajo “working errors” (RMWE), número de veces que entra en un brazo en el que ya ha entrado previamente [4]. Valores más altos en este test se interpretan como peor capacidad cognitiva.

El test de Morris Water Maze se desarrolla en una piscina circular dividida en cuatro secciones imaginarias y se coloca una plataforma sumergida, y por ello oculta a la vista de los animales, en un cuadrante que será siempre el mismo. La rata debe aprender la localización de la plataforma en los entrenamientos mediante pistas visuales en su entorno que le permitan orientarse. En el ensayo final se anota el tiempo que tarda la rata en alcanzar la plataforma, por lo que un mejor aprendizaje en los entrenamientos conllevaría un menor tiempo [4]. Por tanto, valores más altos en este test suponen una peor capacidad cognitiva.

El funcionamiento del test Beam Walking consiste en hacer pasar a los animales 3 veces por un listón, elevado con respecto al suelo, para cruzar entre dos plataformas. Se cuentan las caídas que sufre al cruzar cada una de las 3 veces y se promedian los resultados. Por tanto, un mayor resultado implica una peor capacidad motora.

Por último, el test de Rotarod consiste en anotar el tiempo que se mantiene la rata sobre un cilindro horizontal que rota sobre su eje acelerando su velocidad paulatinamente. Un mayor resultado implica, por tanto, una mejor capacidad motora [5] .

2. Objetivos

Los objetivos principales del presente trabajo se detallan a continuación.

- Análisis integrativo de la información ómica junto con la de los diferentes tests que evalúan capacidades cognitivas y motoras, con la finalidad de poder identificar qué variables ómicas (proteínas o metabolitos) están relacionadas con alteraciones cognitivas o motoras debidas al efecto de alguno de los pesticidas evaluados. Dichas variables ómicas serán candidatas a biomarcadores para el diagnóstico del deterioro de las capacidades neurológicas.

- Determinar los cambios ómicos que se puedan producir en los diferentes sujetos por la acción de los diferentes pesticidas, así como establecer diferencias entre los mismos pesticidas, en el caso de que las haya, en cuanto a sus efectos y/o mayor o menor grado de inocuidad. Esta información serviría para determinar su potencial neurotóxico y ayudaría a determinar el uso de los más convenientes para la salud del consumidor.

Para alcanzar los citados objetivos, se deberán conseguir también los siguientes objetivos secundarios:

- Aplicación de un pre-procesado adecuado tras la exploración inicial de los datos de partida, mediante el uso de herramientas para la corrección de ruido, y la imputación de valores faltantes.

- Pre-selección de variables ómicas de interés atendiendo a su variabilidad entre pesticidas y sexos, para su inclusión en los modelos estadísticos utilizados para el análisis.

- Uso del PLS y *multi-block* PLS como modelos estadísticos multivariantes para la integración de la información ómica y de los tests.

- Selección de variables relevantes en los modelos multivariantes obtenidos.

- Interpretación de los resultados biológicos obtenidos.

3. Materiales y métodos

3.1 Información detallada de las ómicas y los sets

Ómicas

Los datos proteicos llegaron para todos los sets con una cuantificación relativa. Según el protocolo de la empresa encargada de su obtención, Proteome Sciences, las cuantificaciones de los péptidos para cada muestra fueron escalados en función de la mediana, a continuación utilizando estos valores se calcularon ratios, con respecto a las medidas medias de referencia por la tecnología usada, que fueron posteriormente transformados logarítmicamente. Por último la matriz peptídica fue transformada en una matriz proteica. Los valores faltantes (VFs) indicaban que no se obtuvo ninguna señal para la proteína a estudio.

Los datos metabolómicos llegaron cuantificados de forma absoluta, los valores habían sido normalizados por peso mojado (pico más alto dividido por el meso de la muestra a estudio). La tecnología empleada fue: LC-HRTOF-MS y GC-HRTOF-MS, por sus siglas en inglés, *Liquid Chromatography High Resolution Time of Flight Mass Spectrometry* y *Gas Chromatography High Resolution Time of Flight Mass Spectrometry* respectivamente.

Sets

El Set03 constaba de información proteica y resultados de los tests. Los pesticidas y concentración utilizados fueron CAR, CYP, CLOR01, CLOR03 y CLOR1. En la Tabla 1 se muestra información detallada del número de individuos por sexo y tratamiento en los datos proteicos y en los derivados de los tests.

Tabla 1. Información del Set03, número de observaciones (ratas) por tratamiento y sexo para la obtención de los datos proteicos y de los tests. Las observaciones del estudio de proteómica son las mismas para ambos tejidos estudiados, Cerebelo e Hipocampo; T.S. (total por sexo).

Set03	**Observaciones Proteómica: CB &HP**		**Observaciones Tests**	
Tratamiento	**Machos**	**Hembras**	**Machos**	**Hembras**
VH	2	2	3	2
CAR	3	2	3	2
CYP	1	0	1	0
CLOR01	4	1	4	1
CLOR03	4	4	4	4
CLOR1	2	2	2	2
T.S.	16	11	17	11
Total	**27**		**28**	

El Set01 constaba de información proteica, metabólica y derivada de los tests. Los pesticidas utilizados fueron Cypermetrín y Endosulfán. En la Tabla 2 se muestra información detallada del número de individuos por sexo y tratamiento en los datos proteicos, metabólicos y en los derivados de los tests.

Tabla 2. Información del Set01, número de observaciones (ratas) por tratamiento y sexo para la obtención de los datos proteicos, metabólicos y de los tests.

Set01	Observaciones Proteómica:CB		Observaciones Metabolómica:CB		Observaciones Tests	
Tratamiento	**Machos**	**Hembras**	**Machos**	**Hembras**	**Machos**	**Hembras**
VH	3	4	3	4	3	4
CYP	4	4	4	4	4	4
END	4	6	4	6	4	6
T.S.	11	14	11	14	11	14
Total	**25**		**25**		**25**	

3.2 Soporte informático: lenguaje y programa.

El elevado tamaño de los datos ómicos hace indispensable el uso de herramientas informáticas y estadísticas de gran poder para su análisis. Atendiendo a dicha necesidad, se ha utilizado el lenguaje de programación R [6] mediante el interfaz gráfico de RStudio. Los diferentes paquetes de funciones utilizados a lo largo del proyecto se descargaron de los repositorios CRAN[7] y Bioconductor [8], estando este último enfocado al campo de la bioinformática, mientras que CRAN ofrece una variedad más amplia de paquetes, aplicables a estudios tanto biológicos como no biológicos.

El sistema operativo del equipo informático utilizado constaba de una distribución de GNU/Linux Xubuntu 16.04.

En el apartado 8, Adjuntos, se muestran ejemplos de los códigos que se elaboraron, y emplearon, para llevar a cabo el análisis de los datos.

3.3 Métodos estadísticos

3.3.1 Tratamiento e imputación de valores faltantes

El primer paso en el proyecto consistió en el tratamiento e imputación de los VFs generados durante la obtención de los datos. Un valor faltante (VF) es asignado cuando debido a factores ambientales o técnicos no se ha podido anotar el valor de una observación para una determinada variable. Esto no puede considerarse como un valor de 0 para dicha observación ya que asignar el valor 0 supone una cuantificación exacta y objetiva, que se podría interpretar erróneamente como una ausencia de la variable que se pretende medir en la condición estudiada, por ejemplo si se estuviese cuantificando el nivel de una proteína. En concreto tenían VFs los datos de proteómica y los datos de los tests, mientras que los datos de metabolómica estaban completos.

En primer lugar, se observó el número de valores faltantes por variable (de tests o de las ómicas) a lo largo de todas las observaciones (ratas), para descartar aquellas variables con una gran cantidad de los mismos. Se estudió cuantas variables seguirían disponibles para el análisis si se eliminaban aquellas que poseían más de un cierto umbral de VFs, en tanto por ciento.

En segundo lugar y tras realizar el descarte de variables con demasiados VFs, se llevó a cabo la imputación de los VFs restantes. La imputación de datos faltantes se basa en la idea de que cualquier observación en una muestra a estudio puede ser reemplazada por una nueva seleccionada aleatoriamente de la población a la que pertenece la muestra. De esta forma la imputación de un VF es el reemplazo del VF por una estimación obtenida a partir de la distribución de la variable en cuestión [9], de tal forma que los valores imputados no alteren las distribuciones originales de los datos sin imputar. Para llevar a cabo las imputaciones se utilizó el paquete *mice* del repositorio CRAN. Este método fue diseñado para llevar a cabo múltiples

imputaciones en valores faltantes multivariantes, reflejando así la aleatoriedad de los mismos, y utilizando para ello ecuaciones encadenadas. De entre las opciones disponibles en *mice*, se aplicó el método "norm.predict", que reemplaza el VF con el valor predicho por un modelo de regresión lineal obtenido a partir de los datos observados.

3.3.2 Análisis exploratorio y corrección del ruido.

Análisis de Componentes Principales (PCA)

El PCA se utilizó como parte de la exploración de los datos ómicos ya que, al ser un método multivariante de reducción de la dimensión, permite explorar de manera eficiente este tipo de datos que incluyen cientos (o incluso miles) de variables (proteínas o metabolitos).

El PCA es un algoritmo matemático que permite reducir el número de variables (la dimensión) en un estudio mediante la creación de unas nuevas variables latentes, también llamadas componentes, reteniendo la mayor parte de la variabilidad presente en la matriz de datos original [10] . Cada componente principal (PC) es una combinación lineal de las variables originales y explica un porcentaje de la varianza de los datos, de tal forma que la primera componente principal (PC1) explica más variabilidad que la segunda (PC2) y así sucesivamente, y es suficiente elegir un número reducido de dichas componentes para explicar la mayor parte de la variabilidad original. El peso de las variables originales en cada una de las PCs es el *loading* de las variables en dicha PC. Así mismo, se obtiene también el peso de las observaciones en la creación de las nuevas componentes. Dichos pesos se llaman *scores*. Se pueden representar gráficamente las proyecciones de las variables o de las observaciones en las nuevas componentes y este tipo de gráficos ayudan a comprender mejor la relación entre ellas y sirve como control de calidad para comprobar que se agrupan las observaciones correspondientes a los mismos grupos experimentales, o si la forma en que se separan dichos grupos es consistente con el comportamiento esperado.

En los métodos multivariantes como el PCA, para obtener una representación no sesgada de los datos, es a veces necesario centrar y/o escalar las variables para que todas ellas tengan la misma media y/o similar variabilidad. Por norma general, es bastante común centrar los datos, mientras que el escalado suele aplicarse solamente cuando las variables están medidas en distintas unidades.

Para obtener los distintos PCAs se utilizó el paquete *mixOmics* del repositorio CRAN.

Reducción del ruido

Una de las particularidades que ocasiona que el análisis de datos ómicos a gran escala sea especialmente complejo es el elevado nivel de ruido técnico [11]. Este ruido técnico, o variación desconocida ligada a las herramientas de medida y/o al diseño experimental, es recomendable eliminarlo, ya que dificultan los análisis y enmascara los efectos que realmente se quieren estudiar. Para ello se han desarrollado métodos como el ARSyN. Mediante la combinación de análisis de varianza (ANOVA) y la posterior aplicación de análisis multivariantes (*Simultaneous Component Analysis*, SCA) sobre la descomposición de efectos del ANOVA, el método ARSyN es capaz de identificar el ruido presente en los datos y corregirlo [12].

En los datos ómicos de este proyecto se utilizó la función ARSyNseq() del paquete de R *NOISeq* [13] (en Bioconductor) para filtrar el ruido considerando el diseño experimental.

3.3.3 Filtrado de variables con baja variabilidad

Uno de los principales objetivos en los estudios a gran escala, con un elevado número de variables, es reducir el número de las mismas, por diversos motivos. Por un lado, cuanto mayor sea el total de variables, proteínas por ejemplo, más complicado será el posterior análisis de los resultados. Por otro lado, aquellas variables que no varían de forma significativa a lo

largo de las condiciones o tratamientos estudiados no van a ser relevantes y su presencia va a dificultar también los resultados enmascarándolos, en forma de ruido. A lo largo de este proyecto se aplicaron dos estrategias diferentes pero complementarias para tal fin, el filtrado por coeficiente de variación y por expresión diferencial, utilizadas para escoger las variables ómicas con una mayor variabilidad, con las que realizar el análisis multivariante, PLS, el que fue el siguiente paso en el estudio de los datos.

Filtrado por coeficiente de variación

El coeficiente de variación (CV) es una medida de la dispersión de un conjunto de datos que no depende de las unidades de medida de la variable estudiada. Se calcula dividendo la desviación típica de los datos por su media. Dado que el objetivo en este caso era medir la variabilidad entre condiciones experimentales, se calculó el coeficiente de variación de la siguiente forma. En primer lugar se promedió el nivel de expresión para cada variable ómica por tratamiento y sexo. A continuación, se obtuvo la media y la desviación típica de dichos promedios por tratamiento y sexo y, a partir de ellas, se obtuvo el coeficiente de variación. Se seleccionaron para su posterior análisis, las variables ómicas con CV superior a 3.

Filtrado por expresión diferencial

La búsqueda de compuestos diferencialmente expresados (D.E.) entre condiciones es una parte esencial en la comprensión de las variaciones fenotípicas a nivel molecular [14].

En este trabajo, el análisis de expresión diferencial se realizó mediante el paquete *limma* [15], del repositorio Bioconductor. Este paquete utiliza métodos de regresión lineal para modelar la expresión en función de múltiples factores. Además aplica métodos bayesianos para estimar la variabilidad de cada variable ómica y condición con mayor robustez, ya que el tamaño muestral suele ser reducido en este tipo de datos [16] . En este proyecto se estudiaron mediante *limma* los efectos del sexo, del pesticida y la interacción entre ambos. Para poder utilizar modelos de regresión lineal, los datos deben cumplir una serie de requisitos, como seguir una distribución normal y que la varianza de la variable predicha en el modelo sea homogénea en cada valor de la variable predictora. En el caso de que la distribución de los datos no cumpla con dichos requisitos se pueden aplicar transformaciones adecuadas, como la transformación Voom [17].

Las variables seleccionadas para su posterior análisis fueron aquellas sobre las que el pesticida o la interacción tuvieron un efecto significativo (p-valor < 0.05), ya que en este estudio no interesaba analizar las que sólo cambiaran en función del sexo.

3.3.4 Modelos multivariante para relacionar las variables ómicas con las variables cognitivas y motoras

En el modelo multivariante PCA, el objetivo perseguido es reducir el número de variables que describen la varianza de los datos, habiendo tan solo un tipo de variable (p. ej. proteómica). Cuando se manejan diferentes tipos de datos (p .ej. proteómica y datos derivados de los tests), el fin común va a ser la reducción también de la dimensionalidad en cada tipo de datos, pero al mismo tiempo que maximizar la relación de los distintos tipos de variables, es decir, la covarianza. Esto supone una integración de toda la información y puede permitir aumentar la información aportada por los diferentes tipos de variables de forma aislada. Las observaciones deben ser las mismas en cada conjunto de datos.

Partial Least Squares Regression (PLS)

El nombre de PLS hace referencia a una clase de métodos utilizados para relacionar bloques de variables medidas en un conjunto de objetos u observaciones[18], mediante un modelo lineal multivariante. Una de sus principales utilidades deriva de su capacidad para

analizar datos con variables con una elevada colinealidad, ruidosas e incluso incompletas [19]. La modelización mediante PLS tiene un gran potencial como método de análisis en muchas ramas de la ciencia. Tiene su importancia en el campo de la física, de la química, de la química clínica, del control de procesos industriales [20] e incluso en economía.

El PLS se puede considerar un método de predicción. Requiere dos matrices de datos. Una de estas matrices es la matriz de predictores o descriptiva y la otra la matriz de respuesta. El objetivo que persigue el modelo es ser capaz de predecir de la mejor forma posible la matriz considerada como respuesta a partir de la matriz de predictores atendiendo a la estructura o comportamiento común de todas las variables de ambas matrices. Para ello, a la hora de obtener las variables latentes o componentes de las que se ha hablado en el PCA, lo que se busca es que la varianza explicada por cada uno de los componentes sea la máxima posible atendiendo a la localización en el espacio de las observaciones en las dos matrices, maximizando así la covarianza entre ambas.

En este trabajo se aplicó el método del PLS a los datos del Set03, con la matriz de datos de proteómica como matriz de predictores y la matriz de resultados de los tests (cognitivos y motores) como matriz de respuesta, ya que lo lógico es intuir que una posible alteración cognitiva o motora vendrá causada por una alteración proteica, ocasionando dicha alteración proteica un posible retraso o mejora cognitiva y/o motora y afectando por ello al resultado de los tests.

Para aplicar el PLS se utilizó la función pls() del paquete *mixOmics*, del repositorio CRAN.

Multi-block PLS

El *multi-block* PLS es una variante del método del PLS que acepta múltiples bloques descriptivos medidos en las mismas observaciones [21], es decir, que acepta múltiples matrices de predictores.

En este proyecto se utilizaron como matrices de predictores los datos de proteómica y metabolómica del Set01, y como matriz de respuesta los correspondientes datos de los tests.

El objetivo que persigue el modelo es predecir el comportamiento de las variables respuesta a partir de las variables predictoras, mediante el estudio del comportamiento o distribución de todas las variables en el espacio, siendo esto una analogía perfecta con el PLS, con la particularidad de poseer en este caso 2 bloques descriptivos. Por ello el modelo integrará la información de 3 bloques distintos. Para aplicar el *multi-block* PLS (mediante la función block.pls() del paquete *mixOmics*) había que indicar el grado de correlación entre los distintos bloques descriptivos. Por tratarse en este estudio de datos ómicos plenamente relacionados entre ellos, como son la proteómica y la metabolómica, se indicó una correlación máxima de 1.

3.3.5 Selección de variables relevantes en el modelo multivariante

Sparse PLS (sPLS)

Los avances en biotecnología han permitido recolectar una gran cantidad de datos ómicos, siendo uno de los principales problemas derivados de ellos la selección de variables de mayor relevancia de entre la gran cantidad de información. Una opción para abordar este problema es el llamado *sparse* PLS (sPLS). El sPLS combina las buenas propiedades del PLS con el problema de selección de variables [22] teniendo pues como objetivo conseguir de forma simultánea un modelo con una buena capacidad de predicción al mismo tiempo selecciona las variables más importantes para el modelo [23]. La selección de variables se realiza mediante la aplicación de la penalización de Lasso a los *loading vectors* cuando se lleva a cabo la descomposición en valores singulares [22] de las matrices de datos. Con dicha penalización, se

consigue que el número de *loadings* distintos de cero sea mínimo y, por tanto, se excluyen del modelo todas las variables con *loading* cero, conservando las más relevantes.

En este trabajo se ha utilizado la función spls() del paquete *mixOmics* para poder llevar a cabo el sPLS. Esta función requiere que se indique el número de variables que debe retener el modelo por cada componente y por cada matriz de datos al aplicar la selección de variables. Más adelante se detallará cómo se determinó el número óptimo de variables a retener.

Medida del error de predicción: MSEP

El error de predicción cuadrático medio o MSEP (por sus siglas en inglés, *Mean Square Error Prediction*) es un criterio utilizado para seleccionar variables y para llevar a cabo la validación cruzada de modelos estadísticos [24] . Este error se calcula al comparar un valor predicho por el modelo ($\hat{\boldsymbol{y}}$) con el valor real observado ($\boldsymbol{y}$), promediando el cuadrado de la diferencia entre ambos, según la ecuación:

$$\boldsymbol{E(\hat{y} - y)^2} \qquad \textbf{(1)}$$

Al introducir en el modelo los datos de las variables respuesta, que en este estudio son los datos de los tests, el modelo es capaz de calcular el MSEP para cada una de las variables respuesta, ya que puede comparar los valores esperados por la capacidad de predicción del modelo con los valores reales introducidos en él. Valores más bajos del MSEP implicarán una mejor capacidad de predicción del modelo.

El MSEP se utilizó en este proyecto de fin de grado como herramienta para determinar el número de variables ómicas que debía mantener en cada componente el sPLS a la hora de realizar la selección de las mismas. El objetivo perseguido era reducir al máximo el número de variables ómicas seleccionadas por componente al aplicar el sPLS al mismo tiempo que se disminuía (o se aumentaba lo menos posible) el MSEP para cada uno de los tests, en comparación con los MSEP para cada test del PLS. Se realizó de la siguiente forma:

- En primer lugar se calculó el MSEP para cada test en el PLS, es decir, en el modelo inicial con todas las variables ómicas seleccionadas tras el proceso de filtración por coeficiente de variación y expresión diferencial.

- En segundo lugar, y atendiendo a que se trabajaba con 2 componentes, se estudió una gran cantidad de combinaciones del número de variables a seleccionar en cada componente (p ej. 100 en la primera y 110 en la segunda o 140 en la primera y 100 en la segunda, etc.). Se evaluaron todas las combinaciones de números enteros posibles de 10 en 10 unidades desde el 10 hasta el número total de proteínas incluidas en el modelo PLS. Para cada una de estas combinaciones, el sPLS seleccionó las variables a mantener en los 2 componentes del bloque ómico para dicha combinación. Por ejemplo, volviendo sobre la combinación 100 y 110 se establecía que el modelo debía retener las 100 variables con más relevancia para la componente 1 del bloque ómico y 110 para la componente 2 del mismo bloque. Además se obtuvo el MSEP para cada test de cada uno de los sPLS generados para las distintas combinaciones.

- Por último se comparó gráficamente (tal y como se verá en los resultados) el MSEP para cada test de todos y cada uno de los sPLS y del PLS original con todas las variables y se seleccionó la combinación óptima que cumplía el objetivo perseguido, es decir, que utilizando el menor número posible de variables por componente minimizara el MSEP para cada test en comparación con el PLS original, o lo aumentara mínimamente, ya que un ligero aumento del error en el modelo ligado a una gran reducción del número de variables también era algo que interesaba, por la mayor facilidad que derivaría del posterior análisis biológico. Lo que definitivamente no

interesaba era que al reducir el número de variables aumentara significativamente el MSEP para cada test puesto que estaríamos empeorando notoriamente el poder predictivo, y por ende la calidad del modelo establecido.

Variable Importance in Projection (VIP)

El coeficiente VIP mide la importancia relativa de cada una de las variables predictoras (X) en cada componente del modelo, tanto de la matriz X como de la Y, ya que la segunda se predice a partir de la primera. Así pues, el VIP es una medida ampliamente utilizada para clasificar las variables X según su capacidad para predecir Y.

Es una práctica habitual la combinación de los métodos PLS con VIP para llevar a cabo la selección de variables [25]. Un valor de VIP superior a uno suele ser la norma habitual para seleccionar una variable como relevante. Para la primera componente, el VIP se calcula como los *loadings* de dicha componente elevados al cuadrado. Para las restantes componentes, se tienen también en cuenta los *loadings* de las componentes anteriores y además se ponderan utilizando el cuadrado de la correlación entre estas componentes y las variables respuesta

En el caso de la función block.pls() de *mixOmics*, utilizada para obtener el multi-block PLS, no era posible obtener el MSEP para cada variable respuesta, por lo que se recurrió al VIP para seleccionar aquellas variables ómicas con mayor relevancia en el modelo.

Así pues, se calculó el VIP para cada metabolito y proteína incluidos en el *multi-block* PLS (Set01), y se seleccionaron aquellas variables ómicas con un VIP superior a 1.

3.3.6 Relación de las variables ómicas seleccionadas con el deterioro cognitivo y motor

Una vez realizado el proceso de selección de variables relevantes para los distintos modelos multivariantes obtenidos, se estudió la implicación en los procesos cognitivos y motores de las mismas. Para entender dicha implicación se analizó la correlación entre el comportamiento de las variables ómicas con los tests (variables que medían capacidades cognitivas y motoras) a lo largo de todas las observaciones.

Se calculó el grado de correlación de cada variable ómica a nivel individual, (p. ej. cada proteína) con cada test, y se asoció cada una de ellas al test con el que presentaba una mayor correlación en valor absoluto. Las correlaciones se obtuvieron mediante el coeficiente de correlación de Pearson. El valor del coeficiente puede oscilar entre -1 y 1, significando un valor de 1 una correlación positiva máxima, entre las dos variables comparadas, -1 una correlación negativa máxima y los valores cercano a 0 que no existe relación lineal o que ésta es muy baja entre ambas variables.

Las variables ómicas más correlacionadas con los tests de Beam Walking y Rotarod, que miden capacidades motoras, estarían más relacionadas con alteraciones en dichas capacidades, y las variables ómicas más correlacionadas con los tests Morris Water Maze, y Radial Maze con capacidades cognitivas. Una vez clasificadas las variables ómicas entre cognitivas y motoras, atendiendo al signo de la correlación y a la naturaleza de cada test se correlacionaron positiva y negativamente con la mejora cognitiva y motora. Las variables correlacionadas positivamente con la mejora cognitiva eran aquellas correlacionadas negativamente con el test MWM y Radial Maze (puesto que los individuos tardaron menos tiempo en alcanzar la plataforma y cometieron menos errores en el laberinto al aumentar su valor); y las correlacionadas positivamente con la mejora motora eran las correlacionadas negativamente con el test de Beam Walking y positivamente con el test de Rotarod.

En el Set03 estos estudios de correlación se llevaron a cabo de dos formas, por un lado atendiendo a todas las observaciones conjuntamente y por otro separando las mismas en función

del sexo, realizando por ello el estudio de correlación de las proteínas seleccionadas por el sPLS de forma aislada para machos y hembras. La decisión de llevar a cabo un análisis en paralelo separando las observaciones en función del sexo se tomó debido a los resultados de *limma*, mostrados en la Tabla 6, donde se podía observar la fuerte interacción entre el efecto de los pesticidas y el sexo, mostrando ello que las respuestas de las ratas variaban atendiendo al sexo y a los resultados del PLS.

3.3.7 Análisis de enriquecimiento funcional

Los estudios de enriquecimiento funcional son procedimientos inspirados en los criterios de la biología de sistemas. Estas aproximaciones persiguen examinar directamente el comportamiento de bloques funcionalmente relacionados de variables, como puedan ser genes o proteínas, en lugar de centrarse en estudiarlas a nivel individual [26]. Además, hay que tener en cuenta que extraer la información biológica tampoco es una tarea sencilla, para afrontar dicho problema se ha desarrollado un vocabulario para anotar las funciones de las diferentes variables o entidades biológicas (proteínas, genes…) de forma sistematizada [27] que permita acceder a ellas de una forma rápida y estandarizada. El sistema de anotación de *Gene Ontology* (GO) [28] comulga con dicho fin, estableciendo un sistema organizado alrededor de una estructura jerarquizada, que define una serie de términos descriptivos de las diferentes entidades biológicas en 3 aspectos diferentes: procesos biológicos, función molecular y componentes celulares [27]. De este modo, por ejemplo, a partir de una proteína o de su identificador podemos acceder a los términos GO asociados a la misma, y ver en que procesos biológicos participa, qué funciones desempeña y/o si forma parte de algún componente celular. Así pues, gracias a estos términos GO estandarizados se van a poder encontrar por ejemplo dos proteínas distintas que participan o tienen las mismas funciones. Sacando provecho de esto último, atendiendo a este tipo de anotaciones estandarizadas se han podido desarrollar los análisis de enriquecimiento funcional que persiguen encontrar si hay alguna diferencia en cuanto a la abundancia relativa de una determinada función entre 2 grupos de variables.

En este trabajo la anotación de los términos GO de las proteínas se realizó con la información presente en la base de datos de BioMart para *Rattus Norvegicus*, gracias al paquete de R *biomaRt*, atendiendo a que los identificadores (IDs) de las mismas se correspondían con IDs del repositorio de UniProt [29].

Los análisis de enriquecimiento funcional se realizaron a partir de diversos grupos establecidos mediante la correlación de las distintas variables proteicas y metabólicas con los diferentes tests. En concreto, se aplicó el test exacto de Fisher (unilateral) para comprobar si, por ejemplo, aquellas proteínas relacionadas con el deterioro cognitivo estaban enriquecidas en un determinado término funcional como el transporte de glucosa. Así pues, en este ejemplo, se contrastaría la independencia entre las variables binarias "asociación a deterioro cognitivo" y "transporte de glucosa". Un p-valor inferior al nivel de significación fijado (0.05) indica que existe dependencia entre ambas variables y que, por tanto, las proteínas asociadas a deterioro cognitivo están enriquecidas en la función de transporte de glucosa.

3.3.8 Esquematización del flujo de trabajo del TFG

La Figura 2 representa el esquema del flujo de trabajo desarrollado y seguido para búsqueda de biomarcadores, mediante la integración de datos multiómicos y clínicos (tests) , estrategia perfectamente aplicable para futuros trabajos con un mismo fin. Se puede apreciar en ella que está dividida en 2 bloques, haciendo referencia uno de ellos al pre- procesado y a la exploración inicial de los datos, y el otro al análisis en profundidad mediante los métodos multivariantes para llevar a cabo la integración de la información y poder obtener los biomarcadores para las alteraciones neurológicas.

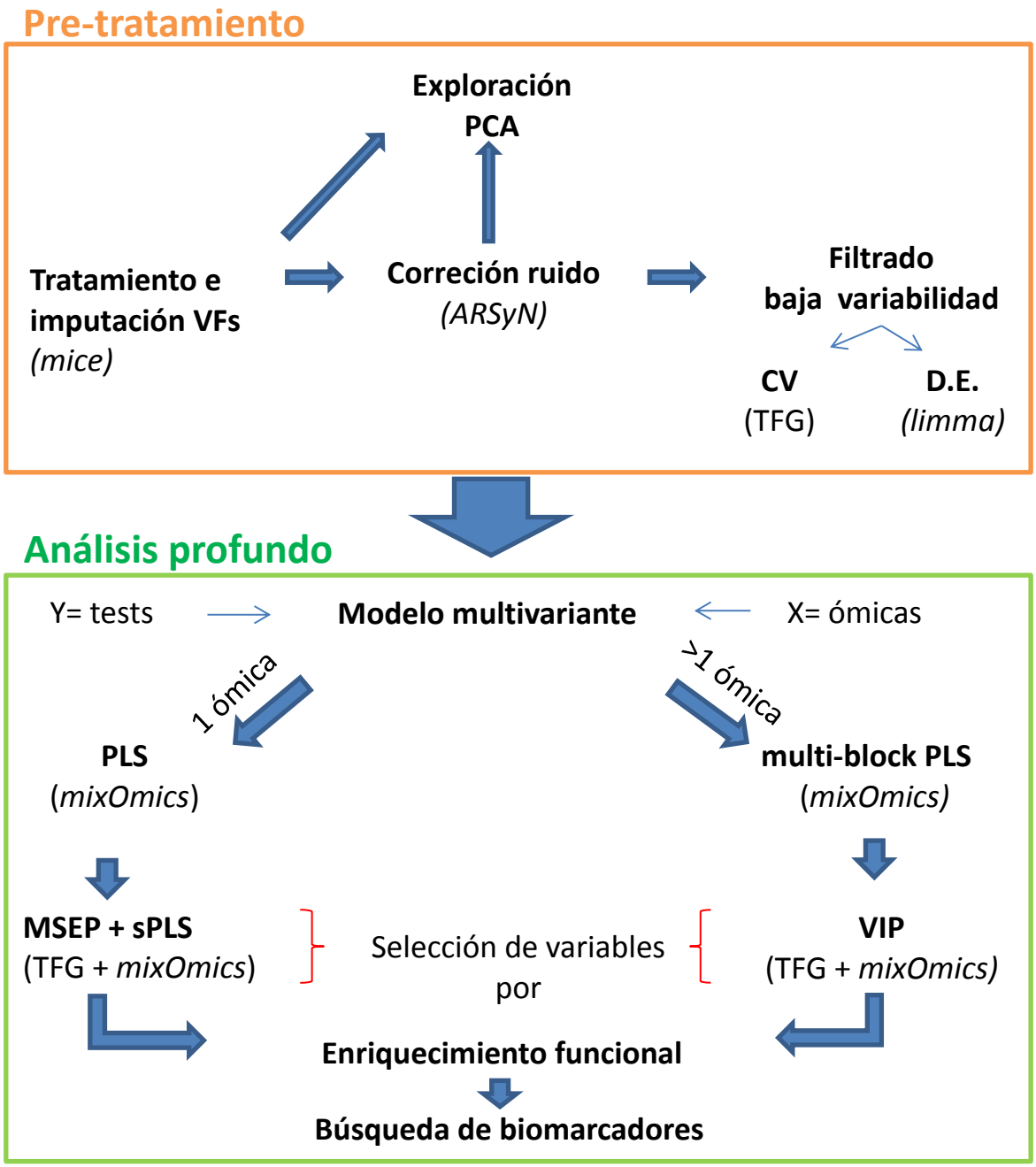


Figura 2. Flujo de trabajo para la interpretación de datos multiómicos y clínicos en la búsqueda de biomarcadores. Entre paréntesis aparecen los paquetes de R utilizados en cada paso. En el caso de aparecer TFG, significa que se desarrollaron funciones propias para poder llevarlo a cabo. X (matriz o matrices de las variables descriptivas), Y (matriz variables respuesta)

4. Resultados

4.1 Resultados Set03

4.1.1 Tratamiento e imputación de valores faltantes

Recordemos que en este conjunto de datos, se disponía de datos de proteómica para comparar los pesticidas Carbaril, Clorpirifos y Cypermetrín con el grupo control.

Al analizar el número de variables y de VFs que quedarían en los datos de proteómica al eliminar aquellas proteínas con más de un tanto por ciento determinado de valores faltantes a lo largo de todas las observaciones (umbral de VFs) se obtuvieron los resultados de la Tabla 3.

Tabla 3. Estudio del número de proteínas y VFs restantes tras eliminar aquellas con un porcentaje de VFs superior a distintos umbrales.

Umbral de VFs	Cerebelo			Hipocampo		
	Nro de Prot. Eliminadas	Nro de VFs Restantes	Nro de Prot. mantenidas	Nro de Prot. Eliminadas	Nro de VFs Restantes	Nro de Prot. mantenidas
25%	2651	6	895	2651	19	895
30%	2651	6	895	2651	19	895
35%	2329	2904	1217	2392	2350	1154
40%	2323	2964	1223	2387	2400	1159

Atendiendo a los escasos valores faltantes restantes tras eliminar todas aquellas variables que tuvieran más de un 25 o 30% de VFs se decidió elegir un umbral menos restrictivo como es el del 40%, es decir, manteniendo todas las variables que no poseyeran más de un 40% de valores faltantes. Así se conservó un número mayor de proteínas, 1223 para cerebelo frente a las 3546 iniciales, con un total de 2964 valores faltantes restantes distribuidos a lo largo de las observaciones, y 1159 para hipocampo, con 2400 valores faltantes.

El estudio de los VFs en los tests para el Set03 mostró los resultados recogidos en la Tabla 4.

Tabla 4. Estudio del número de VFs por test a lo largo de todas las observaciones en el Set03.

Tests	Nº observ. con resultados	Nº observ. sin resultados (VFs)	% VFs	Total Observaciones
MWM	**18**	**10**	**36**	**28**
Rot	**28**	**0**	**0**	
BW	**21**	**7**	**25**	
RMRE	**15**	**13**	**46**	
RMWE	**15**	**13**	**46**	

Se decidió continuar el proyecto sin atender a los datos de los dos tests de Radial Maze del Set03 por poseer estos más de un 40 % de VFs a lo largo de todas las observaciones.

Tras eliminar las variables con un elevado número de VFs se llevó a cabo la imputación de los VFs restantes gracias a la utilización del paquete *mice* del repositorio CRAN.

4.1.2 Análisis exploratorio y corrección de ruido

En primer lugar, se exploraron los datos de proteómica y de los tests mediante PCA. Al realizar los PCAs se centraron todas las variables, y se escalaron únicamente los datos de los tests.

Posteriormente, para reducir el ruido presente en los datos de proteómica se utilizó el método ARSyN. Las Figuras 3 y 4 representan los PCAs para cerebelo e hipocampo, respectivamente, de forma previa y posterior a la corrección del ruido.

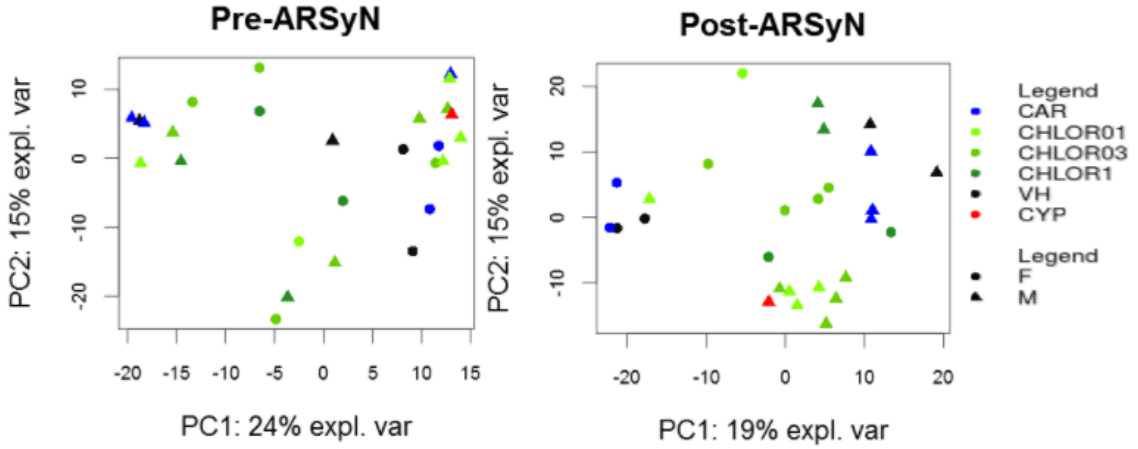


Figura 3. PCA con los datos de proteómica de cerebelo. A la izquierda, antes de la corrección del ruido. A la derecha, tras la corrección. F (hembras), M (machos).

En cerebelo se observó, como fruto de la eliminación de ruido que conllevó la aplicación del ARSyN, que las ratas tratadas con CHLOR tendían a agruparse en torno al centro en la componente principal 1 sin importar el sexo, mientras que los machos controles y tratados con Carbaril se quedaron en un extremo y las hembras controles y tratadas con Carbaril en el opuesto. Esta tendencia se acentuó con el avance del proyecto, tal y como se verá más adelante, y fue de gran importancia en la discusión de los cambios ocasionados por la acción de los pesticidas. La primera conclusión general de este análisis exploratorio fue que, aparentemente, el nivel de expresión de las proteínas de las ratas tratadas con Carbaril era más similar al de los controles que el de Clorpirifos, ya que las ratas tratadas con Carbaril están más cerca de los controles en el PCA. El laboratorio de Neurobiología corroboró que el efecto de Carbaril es más inocuo y que, por tanto, era de esperar que sus resultados se asemejaran más al grupo control.

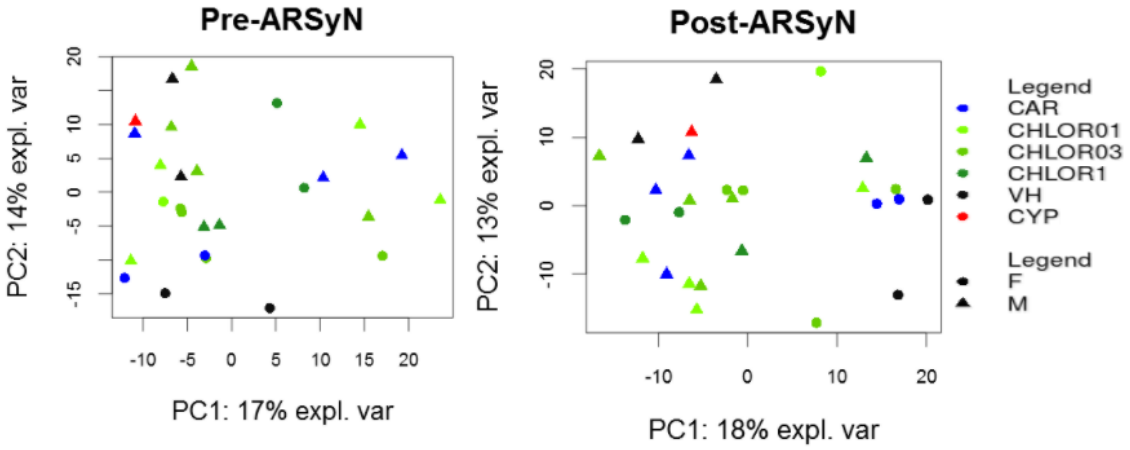


Figura 4. PCA con los datos de proteómica de hipocampo. A la izquierda, antes de la corrección del ruido. A la derecha, de forma posterior. F (hembras), M (machos).

La representación inicial mediante PCA de los datos de hipocampo, sin aplicar la corrección de ruido, no mostró ningún comportamiento de interés ya que las ratas del mismo sexo y tratamiento no se agrupaban bien. La aplicación del ARSyN tampoco aportó ninguna mejora significativa.

Como se puede apreciar en los PCAs que se acaban de mostrar tan solo hay un individuo tratado con Cypermetrín. Es por ello que se decidió dejar de lado esta observación por no poderse contrastar su comportamiento con el de otros individuos de su misma condición. Esto conllevó la disminución en una unidad de las observaciones disponibles para los datos proteicos y de los tests, pasando a tener de este modo 26 ratas con información proteica y 27 ratas con información ligada a los tests.

4.1.3 Filtrado de variables con baja variabilidad

Tras la exploración inicial de los datos, y la corrección del ruido se llevó a cabo el filtrado de variables con baja variabilidad atendiendo al coeficiente de variación de las proteínas y un análisis de expresión diferencial de *limma*.

Resultados del filtrado por coeficiente de variación

Se seleccionaron aquellas proteínas que presentaban un coeficiente de variación mayor a 3, lo que se corresponde con 587 en cerebelo y 477 en hipocampo.

Resultados del filtrado por expresión diferencial

Para realizar el análisis de expresión diferencial se utilizó el paquete de R *limma*, atendiendo al efecto del sexo, de los pesticidas y de la interacción de ambos. Los resultados obtenidos (p-valor <0.05) para cerebelo e hipocampo se muestran en la Tabla 5.

Tabla 5. Resultados de *limma*. Número de proteínas diferencialmente expresadas (D.E.) por condición y por tejido.

limma			
		CB	HP
D.E. por efecto Del pesticida	CHLOR01	35	7
	CHLOR03	23	1
	CHLOR1	30	35
	CAR	1	0
D.E. por efecto De la interacción Pesticida y sexo	CHLOR01	148	68
	CHLOR03	132	0
	CHLOR1	46	65
	CAR	1	1
	Unión	**248**	**112**

Las proteínas diferencialmente expresadas únicamente por efecto del sexo no interesaban, puesto que los cambios causados únicamente por las diferencias entre sexos no eran el objetivo de este estudio, sino los causados por acción de los pesticidas. Por ello no aparecen recogidas en la Tabla 5.

Analizando los resultados de *limma* se pudieron formular diversas hipótesis o predecir diversos comportamientos de lo que estaba sucediendo, los cuales se contrastan a lo largo de este trabajo.

Por un lado, el Carbaryl parecía tener un efecto nulo o al menos mucho menor que el de las diferentes concentraciones de Clorpirifos, de forma global, por las escasas proteínas diferencialmente expresadas atendiendo a su acción.

Por otro lado, el sexo afectaba de forma notoria al efecto de los pesticidas, es decir que la respuesta variaba en función de si eran machos o hembras, tal y como muestran los números de proteínas D.E. mucho mayores al observar los resultados de la interacción entre el pesticida y el sexo frente a los de únicamente los pesticidas.

Otro resultado llamativo surgió de la comparación entre el número total de proteínas diferencialmente expresadas en el cerebelo frente a las de hipocampo, 248 y 112 respectivamente. El total de proteínas en cerebelo es más del doble que el de hipocampo (como ya apuntaban los resultados del PCA), lo que sugirió que el cerebelo fuera más sensible a los cambios que se estaban estudiando, como ya se había constatado previamente en otros trabajos del grupo [30].

4.1.4 Modelo PLS para asociar proteínas con capacidades cognitivas y motoras

Tras filtrar las proteínas atendiendo a su coeficiente de variación y por expresión diferencial se llevó a cabo la unión de ambos conjuntos proteicos para cada tejido, obteniéndose un total de 658 proteínas para el cerebelo y 529 para el hipocampo, tal y como se muestra en la Tabla 6.

Tabla 6. Unión de las proteínas seleccionadas mediante coeficiente de variación (CV) y expresión diferencial para cada tejido.

	CB	HP
CV>3	587	477
***limma* (p.val)< 0.05**	248	112
Unión	**658**	**529**

Los datos de las proteínas correspondientes a dichas uniones fueron los seleccionados para llevar a cabo el PLS. De este modo al aplicar el PLS para el cerebelo se utilizó una matriz proteica con 658 variables y en hipocampo con 529.

De forma previa a la realización del PLS, por la necesidad de presentar en ambas matrices las mismas observaciones se compararon los identificadores de las ratas con datos proteicos y las ratas con datos ligados a las puntuaciones de los tests, para comprobar si eran exactamente las mismas. Tras esta comprobación se tuvo que eliminar una rata de la matriz de datos proteicos, y dos ratas de la matriz de datos de los tests, quedándose de esta forma un total de 25 ratas sobre las que aplicar el PLS. Al introducir la información en el modelo, la matriz con datos proteicos fue la considerada como matriz de predictores y la matriz con datos de los tests, la matriz respuesta.

Los resultados obtenidos tras aplicar el PLS se muestran en las Figuras 5 (cerebelo) y 6 (hipocampo). El número de componentes utilizadas al estudiar el PLS fue de 2. Los datos de los tests para ambos PLS (con los datos proteicos del cerebelo y del hipocampo) son los mismos, puesto que de una rata a la que se le había realizado el test se obtenían los tejidos de cerebelo e hipocampo.

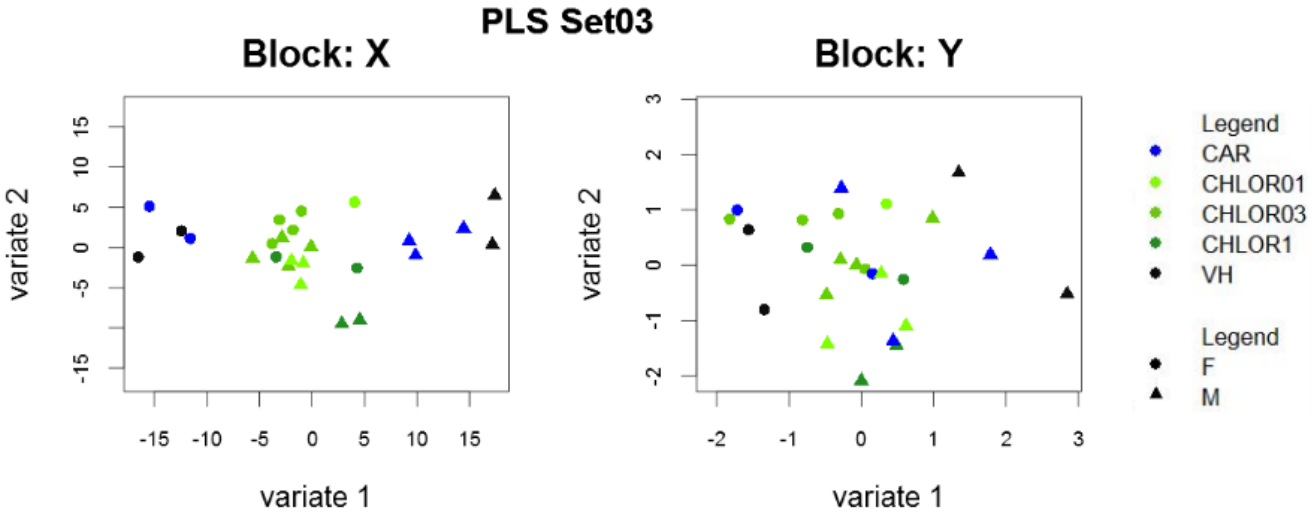


Figura 5. PLS para cerebelo. Block: X se corresponde con la representación gráfica de los *scores* para los datos de proteómica. Block: Y se corresponde con la representación gráfica de los *scores* para los datos de los tests. F (hembras), M (machos).

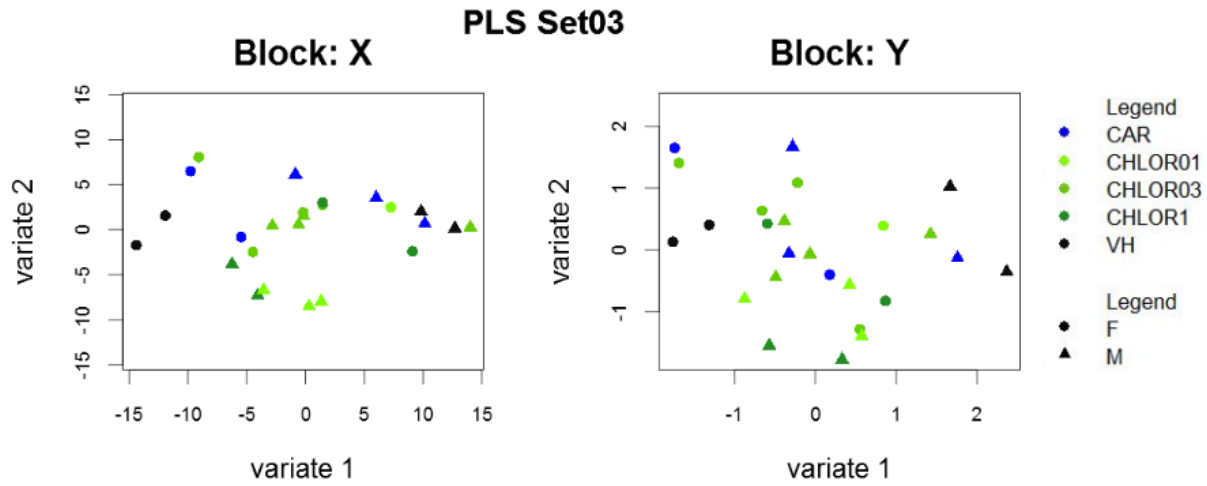


Figura 6. PLS para hipocampo. Block:X se corresponde con la representación gráfica de los *scores* para los datos de proteómica. Block:Y se corresponde con la representación gráfica de los *scores* para los datos de los tests. F (hembras), M (machos).

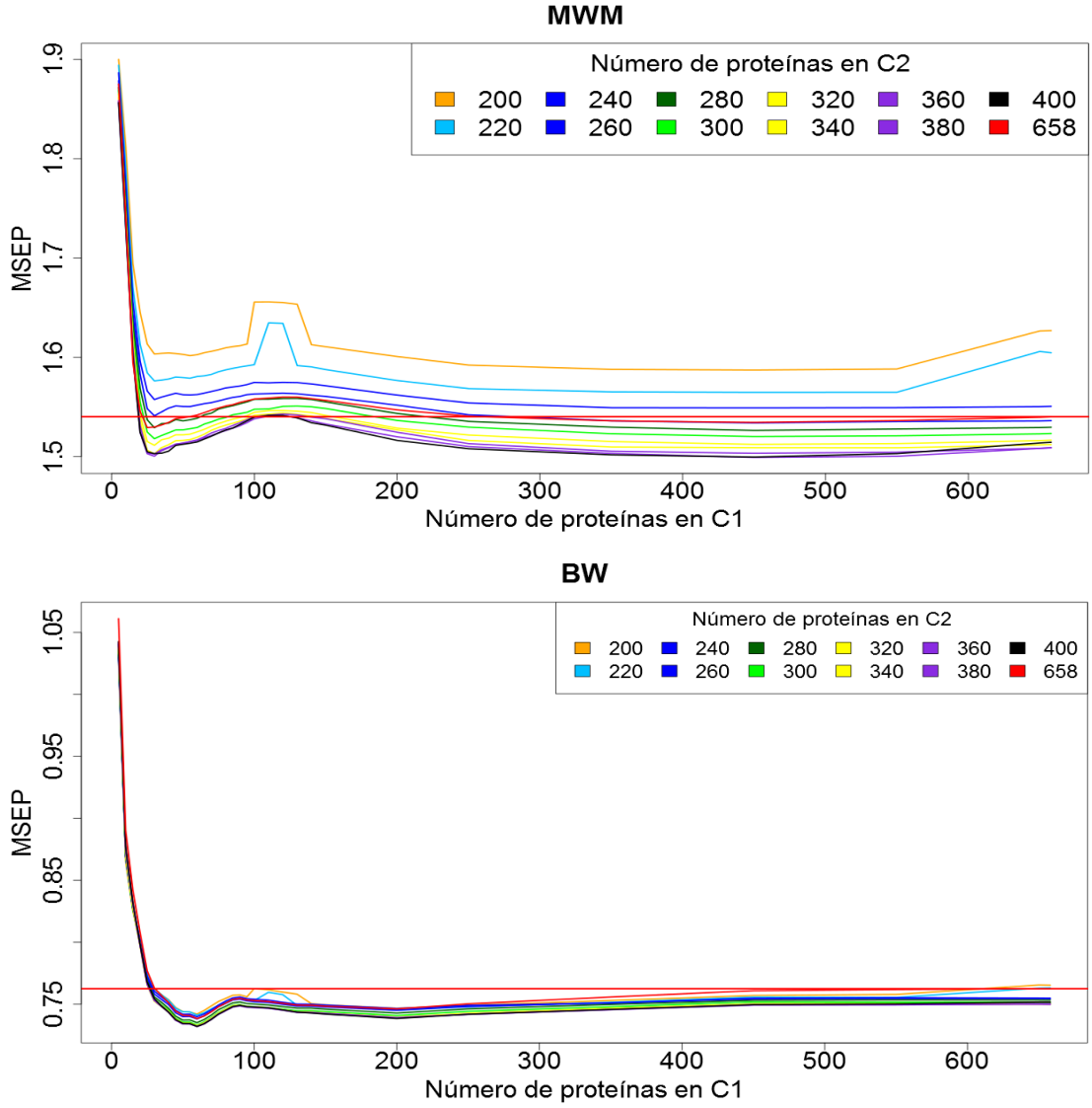
En el Block X del PLS con los datos de cerebelo (Figura 5) se acentuó de forma notoria la tendencia ya observada y comentada en el PCA de las proteínas del cerebelo tras la aplicación de la corrección de ruido. Ahora se observaba claramente, atendiendo a la componente 1 como lo machos tratados con Carbaril se movilizaban junto a los machos control, de forma opuesta a las hembras control, localizadas junto a las hembras tratadas con Carbaril; quedándose tanto los machos como las hembras tratadas con Clorpirifos en el centro. Atendiendo a la componente 2, también cabía destacar el distanciamiento de los machos CHLOR1 del resto de individuos. Se muestra con estos resultados la utilidad del modelo y su capacidad de integración puesto que la información presente en los datos de los tests estaba ayudando a facilitar la comprensión de lo que estaba sucediendo en los datos proteicos, gracias a la mejoría en la separación de las diferentes observaciones.

Como se puede apreciar en la Figura 6, el PLS no mejoró la agrupación de las ratas con los datos de hipocampo. Debido a que los resultados eran tan poco informativos para los datos de hipocampo, pues no permitían establecer diferencias de ningún tipo entre los grupos experimentales, se decidió no profundizar más en el análisis de este tejido. El hecho de que el estudio del cerebelo sí presentara diferencias importantes entre los distintos tratamientos tiene sentido ya que el flujo sanguíneo que recibe el cerebelo es mayor que el recibido por el hipocampo, haciendo de este modo al cerebelo más sensible a cualquier cambio o alteración en el organismo.

4.1.5 Estudio del MSEP para optimizar el número de proteínas a seleccionar

Atendiendo a la capacidad de predicción del PLS y a la posibilidad de estimar el error producido por el modelo al predecir el valor de las diferentes variables respuesta mediante el MSEP, se utilizó dicho error (MSEP) como herramienta para determinar el número óptimo de variables a retener al aplicar el método sPLS, tal y como aparece detallado en el apartado 3.3.6 de Materiales y Métodos.

A continuación se muestran los resultados del MSEP para cada test en función del número de variables (proteínas) seleccionadas por componente al aplicar el sPLS y al aplicar el PLS con los datos originales (658 proteínas), Figura 7. No se muestran todas las combinaciones posibles que se analizaron, solo las más cercanas al número óptimo, por facilitar la visualización e interpretación de los resultados.



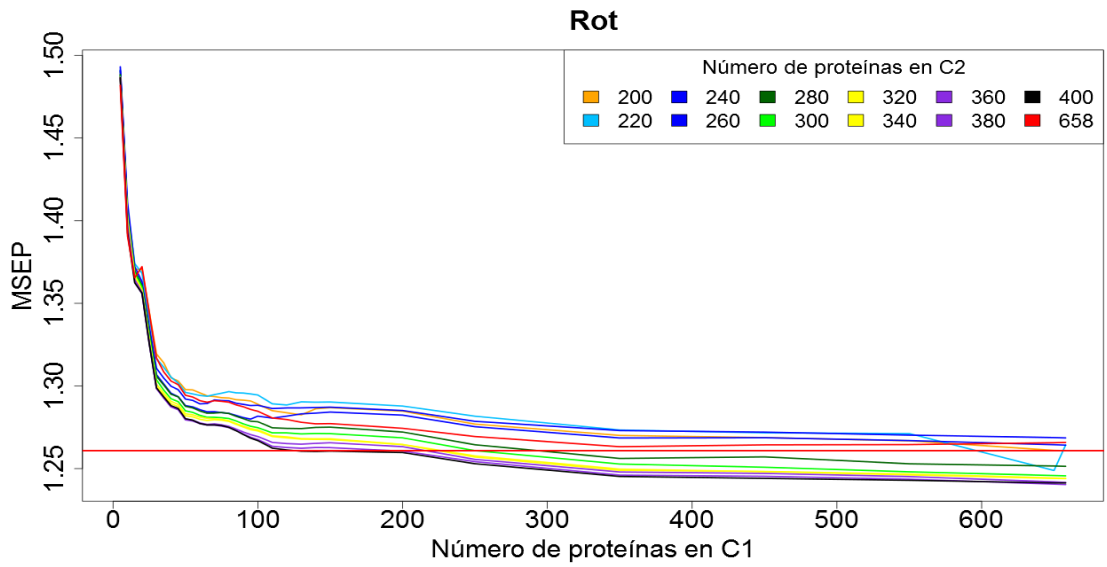


Figura 7. Estudio del MSEP para cada test en función del número de variables seleccionadas por componente al aplicar el sPLS y en el PLS con todas las variables (658). El eje Y representa el MSEP, el X el número de variables seleccionadas en la componente 1. La recta roja horizontal representa el MSEP del PLS para el test estudiado, el resto de líneas representan el número de variables seleccionadas en la componente 2.

Analizando en primer lugar el comportamiento del MSEP en función del número de variables seleccionadas en la componente 1, en los 3 tests se pudo apreciar una reducción notoria del mismo en el intervalo de 20 a 90 proteínas.

Analizando el comportamiento del MSEP en función del número de variables seleccionadas en la componente 2 dentro del intervalo de 20 a 90 proteínas para el componente 1, se observó lo siguiente: en el estudio del test MWM, 300 proteínas era el valor mínimo que minimizaba el MSEP para dicho test frente al del PLS; analizando el BW la selección de 300 proteínas para el C2 minimizaba también el MSEP para el test frente al del PLS. No obstante, en el Rotarod, la selección de 300 proteínas no minimizaba el MSEP frente al PLS aunque la diferencia era escasa. Recopilando estas observaciones y recordando el objetivo buscado en este paso (reducir el MSEP para cada test o aumentarlo mínimamente al reducir el número de variables en comparación con el MSEP para el correspondiente test en el PLS), se decidió aplicar el sPLS manteniendo las 65 variables más relevantes para el modelo en el C1 y 300 para el C2, siendo 65 el mínimo de la línea que representaba 300 variables seleccionadas en el componente 2 para el test de Beam Walking.

4.1.6 Modelo sPLS final

Tras llevar a cabo el estudio del MSEP para cada test en función del número de variables seleccionadas por cada componente al aplicar el sPLS se decidió llevarlo a cabo seleccionando las 65 y 300 proteínas más relevantes para el modelo atendiendo a las componentes 1 y 2 respectivamente. Los resultados gráficos de aplicar dicho sPLS se muestran en la Figura 8.

Tal y como se observa en la Figura 8, al reducir el número de proteínas, la representación de los datos proteicos siguió estableciendo las mismas diferencias observadas previamente en el PLS. Esto reafirmó la utilidad de esta estrategia, puesto que con un número inferior de variables se siguió observando el mismo comportamiento, exactamente con 329 proteínas menos, una reducción del 50% respecto al total de 658 del que se partía al llevar a cabo el PLS calculado al eliminar de las 658 proteínas del PLS las 329 seleccionadas por el sPLS, dichas 329 eran el resultado de la unión del conjunto de proteínas seleccionadas por el C1 (65) del sPLS y el conjunto del C2 (300).

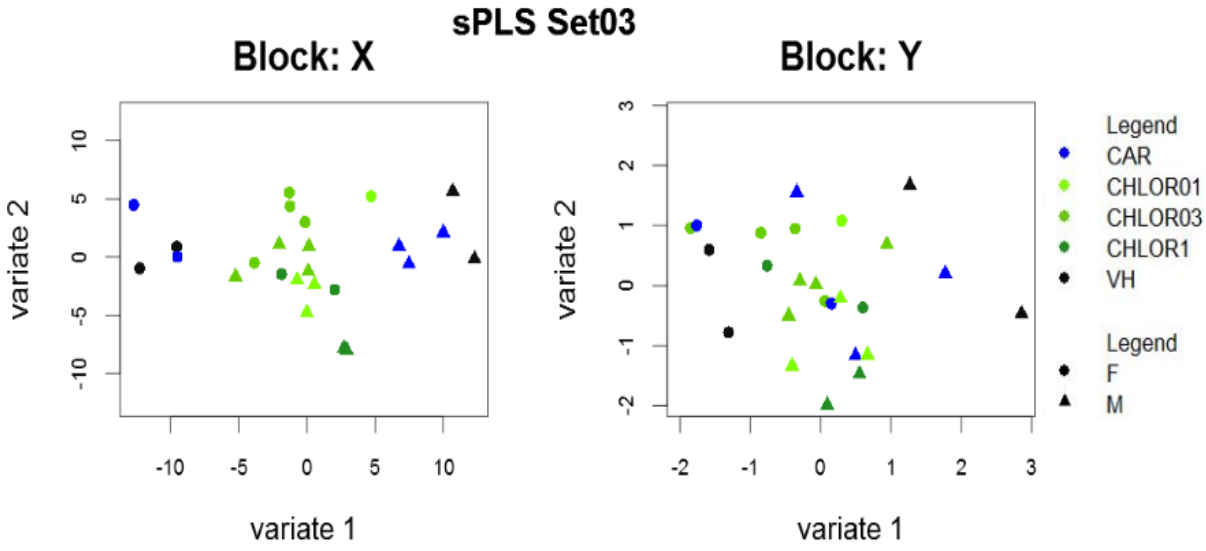


Figura 8. sPLS obtenido al seleccionar 65 proteínas para C1 y 300 para C2. Block:X se corresponde con la representación gráfica de los *scores* para los datos de proteómica. Block:Y se corresponde con la representación gráfica de los *scores* para los datos de los tests. F (hembras), M (machos).

4.1.7 Estudio de la relación entre las proteínas seleccionadas por el sPLS y el deterioro cognitivo y motor.

Para estudiar la implicación de las 329 proteínas seleccionadas por el sPLS con alteraciones cognitivas o motoras se correlacionó el comportamiento de cada proteína con el de cada test a lo largo de todas las observaciones, según aparece detallado en el apartado 3.3.7 de Materiales y Métodos.

A la hora de interpretar los resultados, es muy importante tener en cuenta si el test mide capacidades cognitivas o motoras y qué significa un aumento en la puntuación de cada test, ya que a veces un aumento de la puntuación significa un empeoramiento de la capacidad evaluada. Así pues, para facilitar la interpretación de las correlaciones, hemos considerado que una correlación positiva con la mejora de una capacidad significa que un aumento de la expresión de la proteína estará asociado a una mejora en la capacidad o, equivalentemente, que una disminución de la expresión de la proteína está asociada a un empeoramiento de la capacidad. Para las correlaciones negativas con la mejora de la capacidad, sería justo al contrario. Un aumento del nivel de expresión de la proteína estaría asociado a la pérdida de la capacidad evaluada.

El resultado del número de proteínas correlacionadas con actividades motoras o cognitivas, fue de 201 y 128, respectivamente. El número de las mismas correlacionadas positiva o negativamente con la mejora cognitiva o motora aparece recogido en la Tabla 7.

Tabla 7. Número de proteínas correlacionadas positiva (Corr +) y negativamente (Corr -) con la mejora motora y cognitiva, del total de 329 proteínas seleccionadas por el sPLS.

Corr + mejora motora	102
Corr – mejora motora	99
Corr + mejora cognitiva	62
Corr – mejora cognitiva	66
Total	**329**

En la Tabla 8 se muestra el número de proteínas relacionadas con cada test cuando se separan las observaciones atendiendo al sexo y se estudian de forma aislada entre machos y hembras las correlaciones para las 329 proteínas seleccionadas por el sPLS.

Tabla 8. Número de proteínas correlacionadas positiva y negativamente con cada test en función del sexo.

Capacidad	Machos		
Motora	Beam Walking	Corr +	84
		Corr -	75
Motora	Rotarod	Corr +	22
		Corr -	26
Cognitiva	Morris Water Maze	Corr +	65
		Corr -	57
Capacidad	Hembras		
Motora	Beam Walking	Corr +	34
		Corr -	38
Motora	Rotarod	Corr +	35
		Corr -	59
Cognitiva	Morris Water Maze	Corr +	96
		Corr -	67

4.1.8 Análisis de enriquecimiento funcional.

Para cada uno de los conjuntos de proteínas mostrados en la Tabla 7 se llevó a cabo un análisis de enriquecimiento funcional, cuyos resultados se detallan a continuación.

En el estudio del conjunto de proteínas correlacionadas positivamente con la mejora motora (102) destacaba el proceso de mielinización, por ser de suma importancia para un correcto desarrollo en la actividad del sistema nervioso. La formación de recubrimientos de mielina, generados por oligodendrocitos en el sistema nervioso central es esencial para la propagación y la velocidad de la neurotrasnmisión en los cerebros de mamíferos [31]. Actualmente se ha aceptado que la mielinización es un proceso prolongado que continúa dándose hasta edades avanzadas, por lo que tiene sentido notificar que se estuviesen dando cambios ligados a ella entre las ratas sacrificadas, consideradas como adultos jóvenes, lo que sí que hubiera resultado extraño y menos creíble hubiera sido encontrar diferencias entre ellas en procesos como por ejemplo de neurogénesis que se llevan a cabo fundamentalmente antes de nacer. En [32], se indica que los niveles de mielina se reducen con el envejecimiento, lo que deriva en la reducción de las capacidades cognitivas. Esto se comprende fácilmente puesto que un menor recubrimiento de mielina supone una mayor dificultad para la transmisión de los impulsos nerviosos y por ende del correcto funcionamiento del cerebro. Otro trabajo [33] demostró la alteración de los niveles de mielina en la mente como consecuencia del aprendizaje de habilidades motoras, lo que prueba su plasticidad y la relación entre los niveles de mielina y las capacidades motoras, y refuerza lo observado, es decir, que haya diferencias en cuanto a los niveles de mielina en las ratas con diferentes capacidades motoras.

Analizando las proteínas correlacionadas negativamente con la mejora de las capacidades motoras, 99, aparecían también términos GO relevantes como son: el estrés oxidativo, la respuesta a especies reactivas de oxígeno (ROS) (*Reactive Oxigen Species)*, la respuesta al daño neuronal y a sustancias tóxicas, la transmisión del impulso nervioso y el proceso metabólico del glutatión. El daño axonal y la alteración de la transmisión del impulso nervioso son procesos que lógicamente están ligados con un descenso de las capacidades

motoras. Respecto al estrés oxidativo y la respuesta a ROS, el cerebro es un gran metabolizador de oxígeno (20 % del consumo del cuerpo) [34] con una gran dependencia por ello del mismo para su correcto funcionamiento, aunque posee débiles mecanismos de protección antioxidantes, siendo por ello especialmente sensible al estrés oxidativo y a niveles elevados de ROS. Una gran cantidad de estudios muestran la presencia de estrés oxidativo en numerosas enfermedades con base neurológica o psiquiátrica. Analizando esta susceptibilidad del cerebro por parte del estrés oxidativo es plenamente lógico que el mismo aparezca relacionado con el retraso motor. El glutatión es uno de los principales agentes antioxidantes de nuestro organismo por lo que se comprende que aparezca destacado su metabolismo junto con el de la respuesta celular a ROS, además, alteraciones en los niveles del mismo también se han asociado con enfermedades neuropsíquicas [35]. Y por último, la respuesta a sustancias tóxicas probablemente apareciera por los propios pesticidas, por su efecto nocivo sobre las ratas.

En relación con las proteínas correlacionadas positivamente con la mejora de las capacidades cognitivas no se encontraron resultados dignos de mención. Más interesantes fueron los resultados para las proteínas correlacionadas negativamente. Entre las funciones enriquecidas destacaban: el proceso de envejecimiento, el de aprendizaje o memoria (muy lógico que aparezcan afectados por una disminución de las capacidades cognitivas), y la regulación de la concentración de calcio citosólica. Es bien sabido que con el envejecimiento se ven mermadas o alteradas las capacidades cognitivas, así lo han mostrado muchas investigaciones como [36], por ello no es de extrañar que aparezca dicho proceso ligado al estudio de las proteínas cuyo aumento conlleva un retraso cognitivo. Atendiendo a la alteración de la concentración de los niveles de calcio, el calcio es un mensajero secundario de gran importancia en las neuronas y desempeña un papel fundamental en muchos procesos cognitivos como el aprendizaje y la memoria [37]. Estudios [38] muestran que un aumento de la concentración citosólica del calcio puede ser dañino para la memoria y por ende de los procesos cognitivos.

El pasado 24 de marzo, durante el transcurso de este TFG, una investigadora del laboratorio de Neurobiología del CIPF defendió su tesis doctoral [4], la cual se había desarrollado dentro del proyecto DENAMIC. Aunque en dicha tesis no se analizaron datos ómicos, sí que se estudiaron las variaciones en las puntuaciones medias de los tests (BW, Rot, MWM, RMRE y RMWE) por pesticida y sexo para determinar si había diferencias significativas frente a los controles. Al comparar el comportamiento de los tests de este TFG, con los de la tesis, se observó que en ocasiones eran diferentes. Esto se puede explicar por las grandes diferencias de tamaño muestral entre los 2 estudios. En la tesis se disponía de media por tratamiento y sexo de 13-14 ratas y en los datos de este TFG de 2-3. Por poner un ejemplo, en la tesis había 15 machos tratados con Carbaryl con puntuaciones anotadas para el test BW y en los datos de este trabajo tan solo 3, por lo que se puede afirmar que el tamaño muestral tan reducido en este TFG es el causante de las diferencias en los resultados. No obstante, los resultados eran muy similares para algunos de los tests por lo que centraremos la interpretación biológica de los resultados en el estudio de estos, con el objetivo de identificar las alteraciones biológicas detrás de los cambios en las capacidades cognitivas o motoras, para proponer biomarcadores representativos de dichas alteraciones.

Atendiendo a lo expuesto en el párrafo anterior y a los datos del Set03, y aunque se hizo el análisis para todos los tests, mostraremos aquí los resultados para el test MWM, por presentar un comportamiento muy similar entre ambos estudios. La puntuación promediada del test MWM por tratamiento y sexo con los datos del Set03 aparece representada en la Figura 9. Se observa que la capacidad cognitiva en machos empeora al comparar cada pesticida con el grupo control, especialmente para la máxima dosis de Clorpirifos. En hembras sucede más bien lo contrario: la capacidad cognitiva mejora ligeramente o se mantiene constante, si obviamos la puntuación en CHLOR01 para la que solamente se dispone de una rata y, por tanto, el resultado no es muy fiable. Dada la divergencia de comportamientos en este test para ambos sexos, se van

a analizar los resultados en función de los machos y de las hembras, para tratar de esclarecer que sucede en cada uno de ellos.

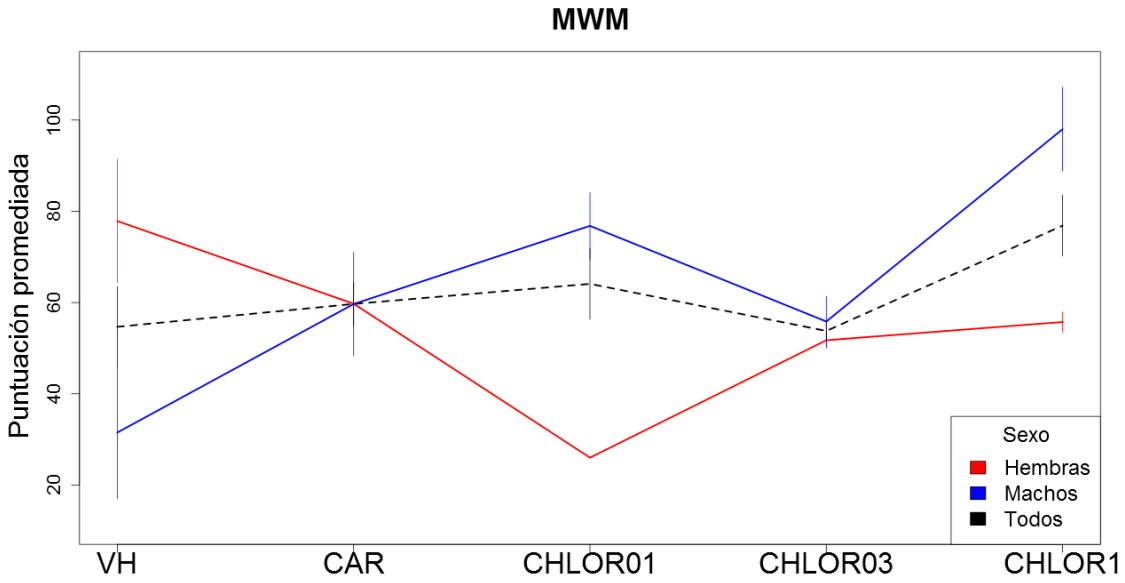


Figura 9. Puntuación media para cada condición del test MWM. Las barras verticales representan el error estándar.

Al analizar las 65 proteínas correlacionadas positivamente con MWM en machos, obtuvimos que estaban enriquecidas en los siguientes procesos: la respuesta al TNF (Factor de Necrosis Tumoral, por sus siglás en inglés), la fosforilación de las proteínas, la participación de las mitocondrias, la regulación de la concentración de los niveles de calcio así como procesos de envejecimiento o procesos de aprendizaje o memoria. Recurriendo a la bibliografía se encontró la siguiente información referente a los términos GO recién comentados. El TNF es una citokina proinflamatoria implicada en procesos de inflamación tempranos [39], en muchas ocasiones se ha estudiado su papel en las alteraciones cognitivas. Por ejemplo, se ha comprobado que muchos trastornos depresivos están asociados con un elevado nivel en el suero y plasma sanguíneo de diversos compuestos, entre ellos el TNF, así como también tratamientos con antagonistas del TNF consiguen mitigar los efectos de la depresión y mejorar los déficits cognitivos ligados a ella [40], [41]. En relación a la fosforilación de proteínas, hay evidencias de que fosforilaciones de algunas de las mismas originan alteraciones cognitivas. Por ejemplo, la proteína Tau, la cual está implicada en muchas enfermedades neurodegenerativas como el Alzehimer [42]–[44] se ha demostrado que su hiperfosforilación va ligada con un descenso de las capacidades cognitivas [45], [46]; su hiperfosforilación puede ocasionar su plegamiento, fragmentación y agregación originando unos depósitos llamados NFTs (*neurofibrilar tangles)* [47]. Respecto a las implicaciones de la alteración de los niveles de calcio ya se ha explicado previamente el impacto dañino que puede tener sobre los procesos cognitivos. Por otro lado, las alteraciones de componentes celulares mitocondriales y por ello de su estructura global puede tener un impacto directo en las capacidades cognitivas de un individuo ya que desempeñan funciones de abastecimiento de energía fundamentales para las neuronas, y gracias al adenosín trifosfato que sintetizan se pueden llevar a cabo reacciones dependientes de energía intracelulares de suma relevancia como la biosíntesis de neurotransmisores [48], [49]. Las mitocondrias son sensibles a las señales extracelulares y esto les puede ocasionar cambios que afecten a los procesos de envejecimientos celulares [50]. Resulta llamativo que pese a que mutaciones en el genoma mitocondrial ocasione en muchas ocasiones enfermedades multi-sistémicas, es generalmente el cerebro el órgano más vulnerable, lo que sugiere una especial sensibilidad por parte de las neuronas a las fluctuaciones bioenergéticas [48].

Respecto a las 96 proteínas correlacionadas positivamente con MWM en hembras resultaron enriquecidos los procesos de mielinización y apareció también el término GO que hace referencia a los oligodendrocitos. Una de las funciones principales de los oligodendrocitos reside en la producción de mielina por lo que tiene pleno sentido la aparición de ambos resultados juntos. Tal y como se ha estudiado y desarrollado en párrafos anteriores, los procesos de mielinización son fundamentales para un correcto funcionamiento del cerebro y del sistema nervioso, teniendo implicaciones en procesos cognitivos y motores. Mientras que al estudiar ambos sexos de forma conjunta los procesos de mielinización aparecían ligados a la mejora motora, al estudiar los sexo por separado aparecieron correlacionados positivamente con el MWM en hembras (que tendía a disminuir levemente, excepto para CHLOR01 donde se producía de forma brusca) y por ello con la mejora cognitiva.

Al analizar los resultados para las proteínas correlacionadas negativamente con el test MWM, en machos, los términos GO enriquecidos fueron bastante inespecíficos y poco relacionados con los procesos estudiados. Para hembras, aparecieron los procesos sinápticos. En [51] se indica que la realización de deporte es capaz de mejorar las capacidades cognitivas mediante su efecto sobre procesos como la sinapsis. En [52] se centran en analizar la plasticidad sináptica y su relación con desórdenes neuropsíquicos. Estas investigaciones atribuyen pues una lógica al hecho de notificar una modulación de los procesos sinápticos ligada a los cambios en las capacidades cognitivas.

El estudio de enriquecimiento funcional de las proteínas asociadas con alteraciones cognitivas y motoras sirvió, pues, para corroborar que las proteínas seleccionadas a partir del modelo PLS eran relevantes para los procesos neurológicos que se pretendían estudiar.

4.1.9. Búsqueda de candidatos a biomarcadores

Con la atención puesta en los cambios en el test de MWM, por las razones ya expuestas, para la búsqueda de biomarcadores se dejó de lado el estudio de las hembras, ya que la estrategia consistía en estudiar aquellas proteínas más correlacionadas con este test, y el mismo mostraba para las hembras, tal y como se ha indicado previamente, un comportamiento anómalo, puesto que apenas cambiaba entre los controles y los tratamiento con: CAR, CHLOR03 y CHLOR1 y en cambio en CHLOR01 se producía una reducción muy brusca, pero como solo había una rata en dicho tratamiento no se podía comprobar la fiabilidad de dicho cambio.

El siguiente paso fue pues analizar de forma individual alguna de aquellas proteínas que presentara una mayor correlación positiva con el test de MWM para los machos, ya que los enriquecimientos funcionales para las correlacionadas negativamente no fueron interesantes, para ver si se podía establecer algún referente o biomarcador de aquellas alteraciones cognitivas de entre las proteínas presentes en este trabajo. De este modo se llegó al análisis de la proteína Anquirina-3 (ANK-3) con el identificador de UniProt F1LPH6, presentando su comportamiento con el del test de MWM una correlación elevada, 0.77, a lo largo de todas las ratos macho tal y como se muestran en la Figura 10 donde se puede observar la gran similitud de su comportamiento, con la del test MWM Figura 9.

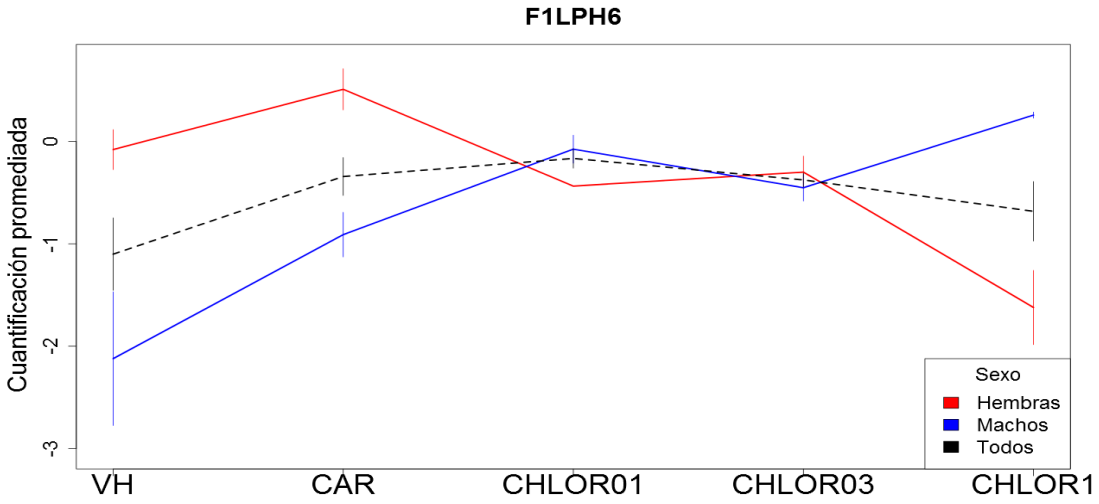


Figura 10. Cuantificación promediada de la proteína con el identificador de UniProt F1LPH6. Las barras verticales representan el error estándar.

Esta proteína, también conocida como Anquirina-G, está codificada por el gen ANK-3, implicado en las capacidades cognitivas. Dada su importancia hay estudios publicados sobre esta proteína. En concreto, en [53], se estudian los efectos cognitivos de las variantes de riesgo del gen ANK-3 en pacientes con trastornos de bipolaridad e individuos sanos. En este artículo se señala que ANK3 influye en el establecimiento de los potenciales de membrana mediante el agrupamiento de los canales de sodio, y desempeña un papel fundamental en la neurotransmisión por lo que debe afectar a los procesos cognitivos, los cuales están afectados en los trastornos bipolares concluyendo que variantes de riesgo del gen ANK-3 pueden tener un impacto en las funciones neurocognitivas, y sugiriendo un mecanismo por el que alteraciones de ANK confieren un riesgo de sufrir un trastorno de bipolaridad. En [54] se relacionaron los niveles de ANK-3 con: el carácter, el estrés y la longevidad, mediante el estudio de muestras sanguíneas de pacientes diagnosticados con desórdenes psiquiátricos. En este estudio se indicó que niveles más altos de ANK3 habían sido hallados en pacientes que se habían suicidado. Y de forma independiente también se habían observado niveles elevados de ANK3 en individuos con Hutchinson-Gilford un síndrome asociado con el envejecimiento acelerado.

Cabe remarcar también, atendiendo a la Figura 10, que los niveles proteicos promediados de F1LPH6, en los machos, son superiores en todas las concentraciones de Clorpirfos frente a las ratas tratadas con CAR, que sería lo esperado por la mayor inocuidad de este pesticida en comparación con el Clorpirifos.

Dado que Clorpirifos era el pesticida con mayor efecto, nos preguntamos cuáles de las 329 proteínas seleccionadas por el sPLS estaban diferencialmente expresadas según *limma* por la acción de las tres concentraciones distintas de Clorpirifos, sin tener en cuenta si afectaban o no de forma diferente en función del sexo. Como respuesta a dicha pregunta se obtuvieron 3 proteínas, los identificadores de UniProt de las cuales eran: D4ADF5, G3V949 y D4A786. La proteína D4A786 no se estudió ya que su entrada en UniProt había quedado obsoleta.

Prestando atención a D4ADF5, esta proteína se encuentra codificada por el gen Pdcd5 (de su nombre en inglés: *Programmed Cell Death protein 5-like*) y su expresión promediada a lo largo de los diferentes tratamientos aparece recogida en la Figura 11.

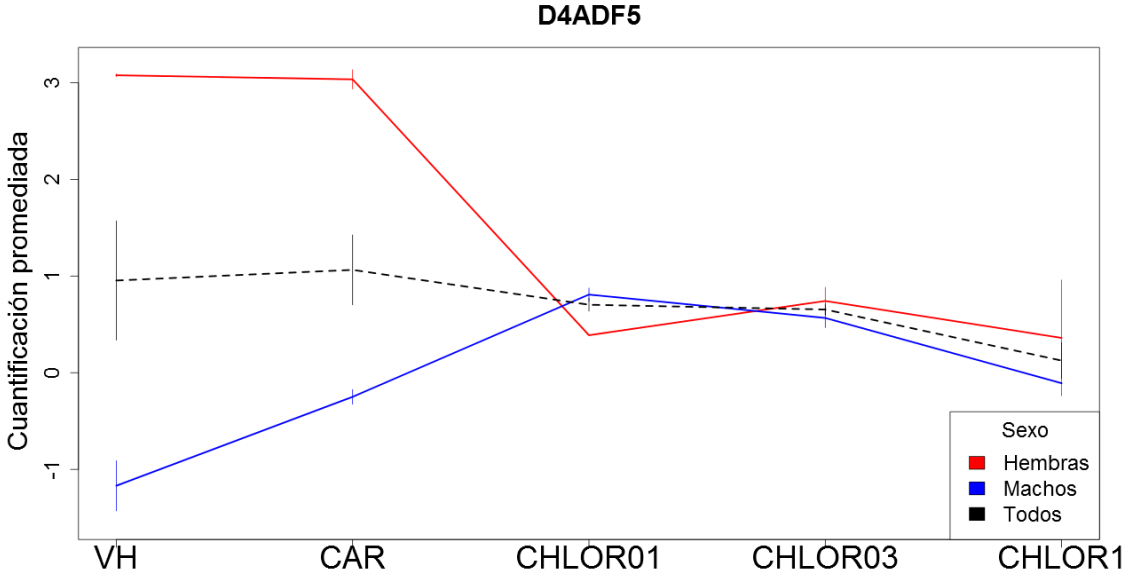


Figura 11. Cuantificación promediada de la proteína con el identificador de UniProt D4ADF5. Las barras verticales representan el error estándar.

Se pudieron observar en la Figura 11 diversos comportamientos a remarcar y de interés.

Por un lado, se corroboró de nuevo lo ya observado en el PLS y el sPLS donde los machos y las hembras tratadas con Carbaril y control (machos por un lado y hembras por el otro) presentaban grandes similitudes entre sí, y mostraban comportamientos opuestos atendiendo a los sexos; sobre todo se veía en este caso como entre las hembras tratadas con Carbaril y control prácticamente no había diferencia alguna

Por otro lado, resultaron llamativas y precisaban de explicación también estas grandes diferencias basales observadas entre los controles machos y hembras neutralizadas al administrar los pesticidas, sobre todo Clorpirifos. Analizando trabajos sobre el gen Pdcd5 fue posible tratar de establecer una hipótesis sobre este comportamiento. En un trabajo con ratas se observó que la expresión de seis genes era significativamente diferente entre sexos, entre ellos se incluía el Pdcd5 cuya expresión era más elevada en hembras comparada con los machos [55] tal y como se observa en la Figura 11. Este dimorfismo sexual explicaría dichas diferencias basales notorias y observadas al estudiar muchas otras proteínas en el Set03. Continuando con el estudio del gen Pdcd5 se encontró otro artículo, [56], en el cual se indicaba que el Clorpirifos altera la expresión de un conjunto de genes, entre ellos el Pdcd5, con diversas funciones en los astrocitos primarios humanos. El mismo artículo indicaba que Clorpirifos aumenta la expresión del gen Pdcd5. Esto explicaría el comportamiento observado en macho pero no en las hembras, donde al parecer la administración del Clorpirifos disminuye la cantidad de la proteína producida por el gen Pdcd5. No obstante, otro trabajo , [57], ayudó a comprender por qué podía estar dándose dicho comportamiento de respuesta diferencial entre machos y hembras, el artículo se centraba en estudiar las diferencias sexuales en la respuesta inflamatoria de los astrocitos y exponía que los astrocitos son uno de los tipos de células gliales que muestra diferencias sexuales en cuestión de número, diferenciación y función; y concluía diciendo que los astrocitos de machos y hembras responden de forma diferente ante un proceso inflamatorio. Es comprensible que los machos y las hembras puedan responder completamente diferente a un mismo estímulo atendiendo a las diferencias existentes a nivel bioquímico entre cada sexo, cobrando por ello sentido también que un pesticida pueda afectar a la capacidad cognitiva de un sexo y no al del otro.

Recopilando la información y los resultados obtenidos hasta el momento se estableció que sucedía lo siguiente: los Clorpirifos neutralizaban las diferencias bioquímicas que existen entre machos y hembras, no así el Carabaryl, o al menos no de forma tan acusada, por lo que se consideró un pesticida mucho más inocuo que el Clorpirfos y más recomendable por tanto su uso. Además se atribuyó un significado biológico a las alteraciones cognitivas causadas en los machos por efecto de las distintas concentraciones de Clorpirifos, notificadas en la tesis [4], seleccionando a la proteína F1LPH6 como un posible biomarcador para dichas alteraciones cognitivas.

4.2 Análisis PLS por sexo

Puesto que en los resultados de los análisis multivariantes se observaron unas diferencias basales entre los controles neutralizadas o reducidas al administrar los pesticidas se decidió analizar los sexos por separado para tratar de identificar proteínas sin diferencias entre sexos en control, cuyos cambios de expresión se debieran principalmente al pesticida. Para ello, se realizaron dos PLS aislados entre ellos, uno con la información de las ratas macho y otro con la de las ratas hembra, obteniéndose los resultados mostrados en las Figuras 12 y 13. Las proteínas de partida sobre las que se aplicó este PLS fueron las 658 que provenían del análisis de filtrado por baja variabilidad ya utilizadas en el anterior PLS con machos y hembras juntos.

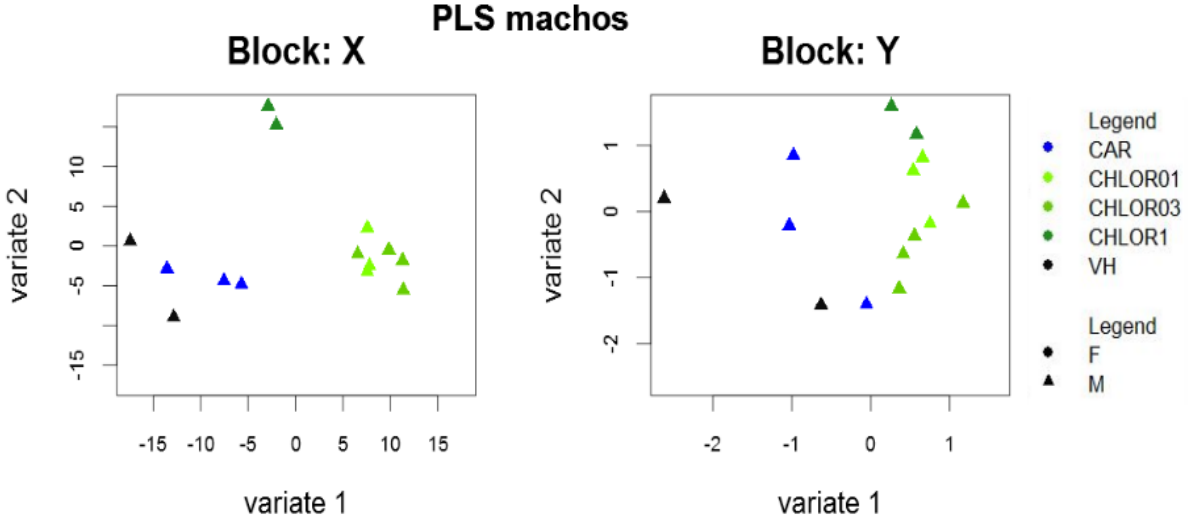


Figura 12. PLS para las ratas macho del Set03 con la información proteómica del cerebelo. Block:X se corresponde con la representación gráfica de los *scores* para los datos de proteómica. Block:Y se corresponde con la representación gráfica de los *scores* para los datos de los tests. F (hembras), M (machos).

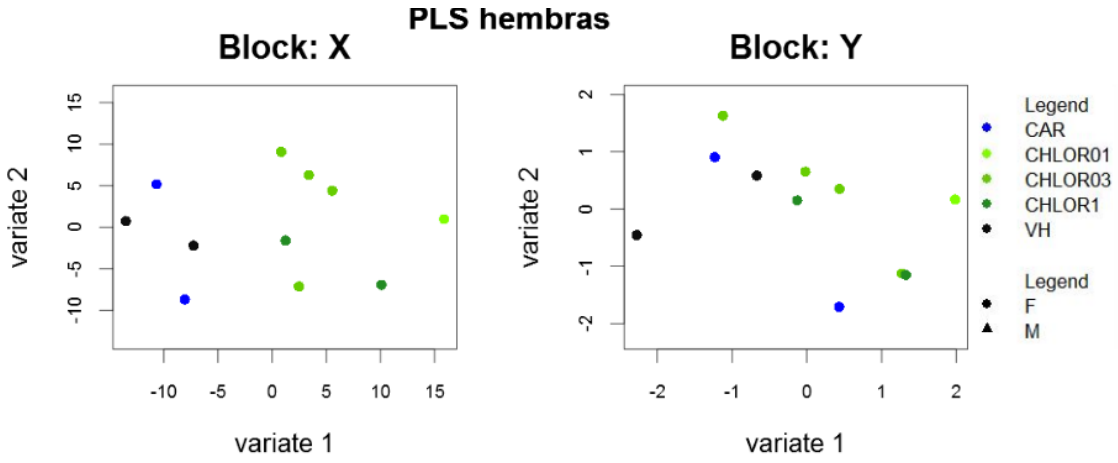


Figura 13. PLS para las ratas hembra del Set03 con la información proteómica del cerebelo. Block:X se corresponde con la representación gráfica de los *scores* para los datos de proteómica. Block:Y se corresponde con la representación gráfica de los *scores* para los datos de los tests. F (hembras), M (machos).

En una primera aproximación, sobre los resultados del PLS para cada sexo, se estableció que el sPLS mantuviese las 100 proteínas más relevantes en cada una de las 2 componentes con las que se estaba trabajando. Así pues el número total de proteínas considerado por el sPLS aplicado sobre las ratas macho fue de 182 (ya que se excluyeron del conteo aquellas proteínas repetidas que eran relevantes para ambas componentes) y en hembras 174. Al comparar dichas 182 y 174 proteínas con las que ya se habían analizado previamente con el sPLS que incluía machos y hembras se vio que la mayoría de ellas ya habían sido incluidas en el modelo anterior probando por tanto la eficacia del mismo, ya que tan solo 42 de las 182 y 53 de las 174 no habían sido identificadas previamente.

Para tratar de encontrar el comportamiento buscado en esta estrategia se realizaron y exploraron visualmente representaciones gráficas promediadas por tratamiento y sexo para cada una de las 42 y 53 proteínas no estudiadas previamente en machos y hembras. Tras llevar a cabo dicho análisis, solo cabe destacar 1 proteína encontrada entre las 42 de los machos, Q64640 (Figura 14).

Tal y como se observa en la Figura 14 la proteína con el identificador de UniProt Q64640 que se corresponde con una adenosina quinasa presentaba esta vez un valor semejante entre los controles, siendo alterado luego por el tratamiento con los pesticidas. En los machos se aprecia una tendencia de aumento al incrementar la concentración de Clorpirifos. En las hembras se aprecia claramente una condición, CLOR01, que no sigue la tendencia al alza observada en machos. El problema con dicha condición, como ya se ha comentado, es que solo hay una observación, por lo que no es posible saber si se trata de un comportamiento anómalo o común. Volviendo de nuevo al estudio de la proteína y de los machos, el aumento de la presencia de esta quinasa tendría mucha relación con el aumento de las fosforilaciones estudiadas anteriormente, que aparecían como resultado de los enriquecimientos funcionales de las proteínas correlacionadas positivamente con el test de MWM en machos.

Es necesario hacer hincapié en que aunque se encontraran diversas proteínas con el comportamiento buscado en este apartado, el comportamiento de la mayoría de las 42 y 53 estudiadas seguía siendo el mismo que el observado al analizar machos y hembras de forma conjunta, es decir que había una diferencia basal grande entre machos y hembras control neutralizada al aplicar los pesticidas, fundamentalmente el Clorpirifos. El laboratorio de Neurobiología constató que era perfectamente normal observar estas diferencias entre sexos en capacidades cognitivas y motoras ya en los sujetos control.

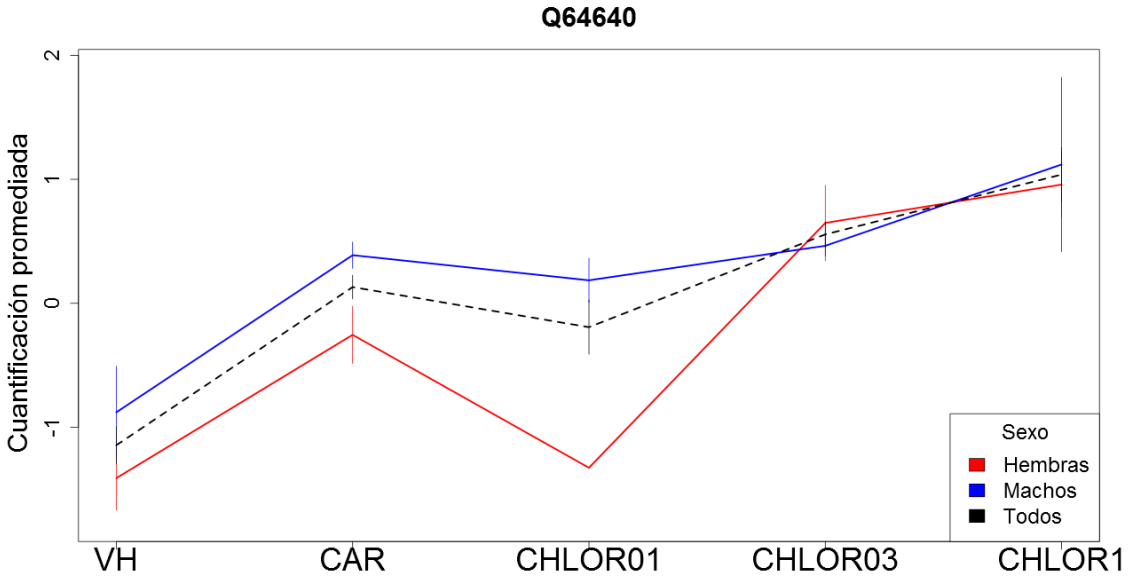


Figura 14. Cuantificación promediada de la proteína con el identificador de UniProt Q64640. Las barras verticales representan el error estándar.

4.3 Resultados Set01

4.3.1 Pretratamiento y exploración inicial de los datos

Los datos ómicos del Set01, 112 metabolitos y 820 proteínas, habían sido utilizados un año antes en otro TFG [30] para llevar a cabo otra aproximación diferente. Así pues ya no fue necesaria la corrección de ruido ni la imputación de VFs, para la proteómica y metabolómica.

Los datos de los tests, en cambio, no habían sido utilizados anteriormente por lo que sí que se precisó una imputación de sus valores faltantes. El número de los mismos para cada test aparece recogido en la Tabla 9.

Atendiendo a los criterios utilizados en el Set03 y puesto que ningún test superaba el 40 por ciento de valores faltantes a lo largo de las 25 observaciones se decidió continuar el estudio con todos ellos. Dichos VFs se imputaron también utilizando el paquete *mice* del repositorio CRAN.

Tabla 9. Estudio del número de VFs por test a lo largo de todas las observaciones (observ.) en el Set01.

Tests	Nro. de observ. con resultados	Nro. de observ. sin resultados (VFs)	% VFs	Total Observaciones
MWM	**17**	**8**	**32**	**25**
Rot	**23**	**2**	**8**	
BW	**19**	**6**	**24**	
RMRE	**23**	**2**	**8**	
RMWE	**23**	**2**	**8**	

A continuación se exploraron los datos de proteómica y metabolómica mediante PCAs, recogidos en la Figura 15.

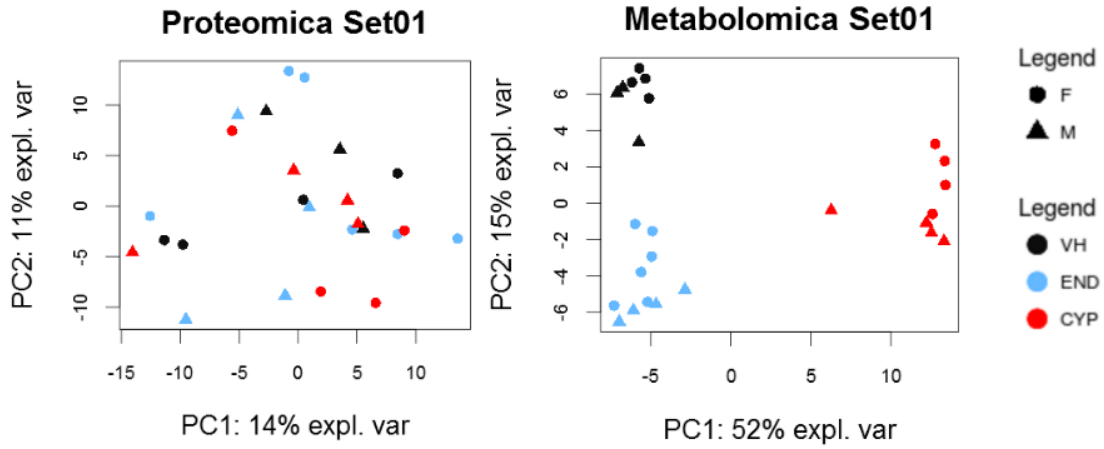


Figura 15. PCA proteómica (izquierda). PCA metabolómica (derecha). F (hembras), M (machos).

En el PCA de los datos proteicos, las observaciones no se agrupan según el tratamiento, sino que están mezcladas entre sí, a pesar de que se había corregido el ruido en los datos. Esto indica una mala calidad de los mismos. No obstante, en el de los datos metabólicos observamos que PC1 separa las ratas tratadas con Cypermetrín frente a las control y a las tratadas con Endosulfán, mientras que la PC2 separa las ratas control de las tratadas Endosulfán.

4.3.2 Resultados *multi-block* PLS

Para el análisis mediante el *multi-block* PLS se utilizaron los datos de las 25 observaciones para los 112 metabolitos, 820 proteínas y 5 tests. Se definieron las matrices de proteómica y metabolómica como las matrices descriptivas o predictoras y la de los tests como la matriz de respuesta. Se indicó al modelo, tal y como se detalla en el apartado 3.3.5 de Materiales y Métodos una correlación entre ómicas de 1, es decir máxima. Los resultados obtenidos se muestran en la Figura 16.

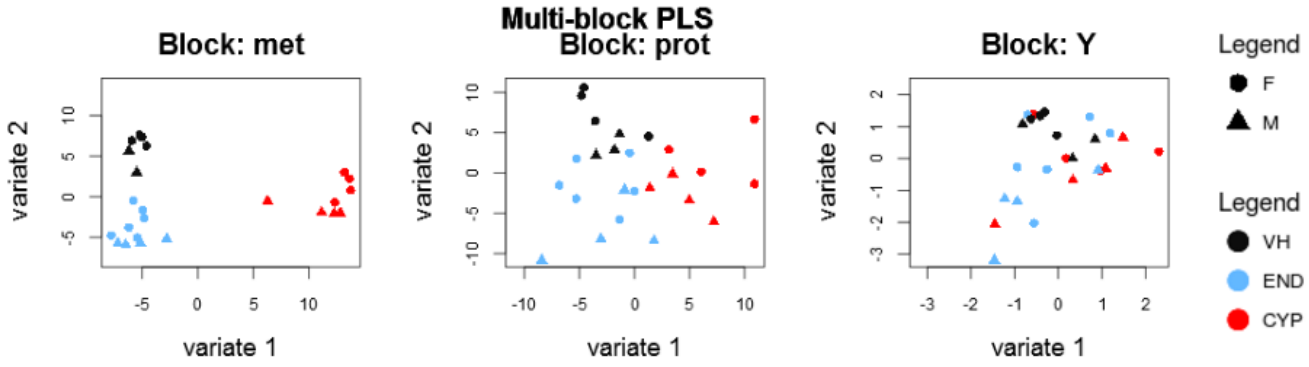


Figura 16. Resultados *multi-Block* PLS. "Block: met" (izquierdo) y "Block: prot" (central) se corresponden con la representación de los *scores* para los datos de metabolómica y proteómica respectivamente. "Block:Y" se corresponde con la representación de los *scores* para los datos de los Tests. F (hembras), M (machos).

Tal y como se puede observar en la Figura 16, gracias a la información aportada al modelo por las distintas matrices, la representación de los scores para los datos de proteómica separa mucho mejor los distintos tratamientos que el PCA. Los datos metabólicos continúan separándose de la misma forma que al explorarlos de forma aislada mediante PCAs, lo que refuerza y da soporte a la gran utilidad y capacidad de integración de este modelo estadístico. Sin embargo, los datos de los tests se presentan más ruidosos, como ya sucedía en el Set03. Esto puede ser debido a una mayor variabilidad biológica en este tipo de mediciones, o bien a la propia dificultad de obtener este tipo de medidas, que conlleva una menor precisión y por tanto hace necesario el incremento del tamaño muestral para obtener resultados fiables, tal como se ha comentado anteriormente.

Analizando el comportamiento de los machos y las hembras dentro de cada tratamiento en la Figura 16 se puede observar también como los machos tienden a estar separados de las hembras fruto de las diferencias biológicas entre sexos. En este caso, la separación de los controles no es tan visible como en los controles del Set03, probablemente porque en el Set01 el efecto de los pesticidas estudiados es mayor que el de los pesticidas del Set03 y esto hace que las diferencias entre sexos no destaquen tanto.

4.3.3 Selección de variables mediante VIP

Como se comentó en Materiales y Métodos, no era posible calcular el MSEP para la selección de variables en la función de *mixOmics* para el *multi-block* PLS. Por ello, se calculó el VIP para cada una de las variables ómicas (120 metabolitos y 820 proteínas), y se seleccionaron aquellas con un VIP mayor a 1, obteniéndose 265 proteínas y 23 metabolitos (Figura 17). Como se puede observar el resultado es muy similar al de la Figura 16 (realizado con los 120 metabolitos y 820 proteínas), lo que confirmó que la estrategia de selección de variables mediante VIP era adecuada, ya que el modelo mostraba los mismos comportamientos con un número de proteínas y metabolitos mucho menor.

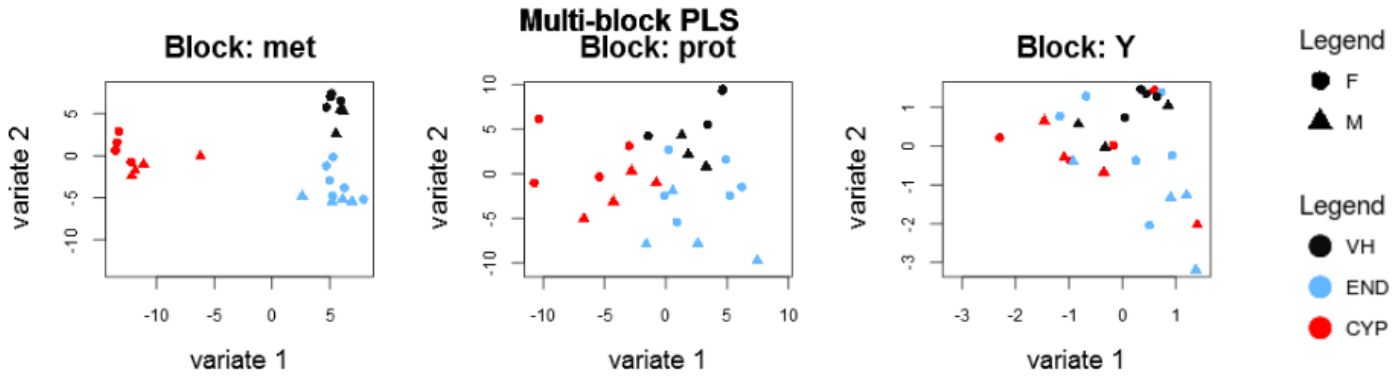


Figura 17. *Multi-block* PLS realizado con 23 metabolitos y 265 proteínas. "Block: met" (izquierdo) y "Block: prot" (central) se corresponden con la representación de los scores para los datos de metabolómica y proteómica respectivamente. "Block: Y" se corresponde con la representación de los scores para los datos de los Tests. F (hembras), M (machos).

4.3.4 Búsqueda de candidatos a biomarcadores

En primer lugar, y mimetizando la estrategia utilizada en el análisis de los datos del Set03, se compararon los comportamientos de los tests en este estudio con los de la tesis [4].En este caso, nos centraremos en el estudio del test RMRE, ya que presentó uno resultados muy similares entre ambos estudios. El objetivo era de nuevo tratar de esclarecer los cambios biológicos detrás de las variaciones en este test y encontrar algún biomarcador representativo de las alteraciones cognitivas ligadas a la variación de los errores de referencia en el test de Radial Maze.

Como muestra la Figura 18, se notificó un aumento de los errores de referencia en este test para los machos tratados con pesticida frente a los controles, es decir, un mayor deterioro cognitivo bajo los efectos de los pesticidas evaluados. En las hembras, el cambio en la puntuación de los tests fue menos importante.

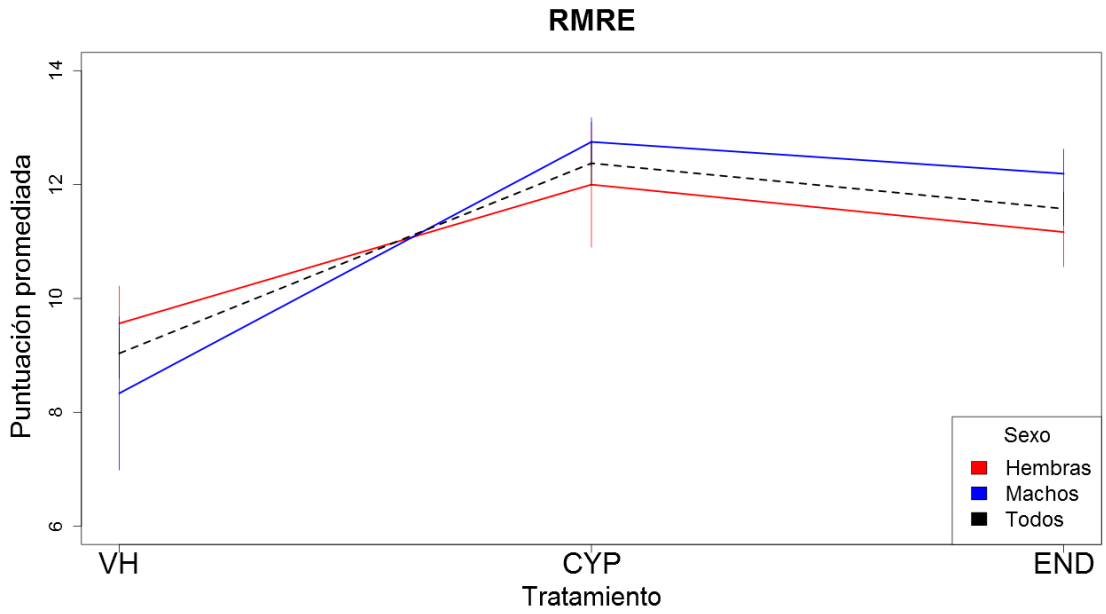


Figura 18. Puntuación media, para cada tratamiento, del test RMRE. Las barras verticales representan el error estándar.

El siguiente paso consistió en correlacionar mediante Pearson el comportamiento de las variables ómicas seleccionadas mediante el VIP (265 proteínas y 23 metabolitos) a lo largo de las observaciones con cada uno de los tests. Posteriormente se agruparon las variables ómicas con aquellos tests cuya correlación fuera mayor en valor

absoluto. Este análisis se realizó en función del sexo, es decir separando previamente los machos de las hembras. Una vez hecho esto se estudiaron las proteínas, 30, y metabolitos, 9, que presentaron una mayor correlación con el test RMRE en machos. No se estudiaron las hembras porque, como se ha comentado anteriormente, las variaciones de puntuación en los tests fueron menores, por lo que es más difícil establecer una correlación consistente con los datos ómicos.

Se descartó llevar a cabo un análisis de enriquecimiento funcional con metabolitos por el número tan reducido de los mismos. Se llevó a cabo para las 30 proteínas, sin obtener ningún resultado destacable.

En machos, se identificaron como relevantes por su comportamiento y por su correlación con el test RMRE dos metabolitos, la ornitina y la guanosina y una proteína, P21707 (identificador UniProt).

Respecto a la guanosina, este metabolito presentó una correlación de 0.7 (en función de los machos) con el test RMRE, y su comportamiento aparece representado en la Figura 19. Se puede observar que dicho comportamiento es muy similar al del RMRE (Figura 18). Por lo que se puede asociar un aumento en la cuantificación del metabolito en ratas tratadas con pesticidas respecto al control, a un mayor deterioro cognitivo. El deterioro cognitivo es más pronunciado en Endosulfán que en Cypermetrin.

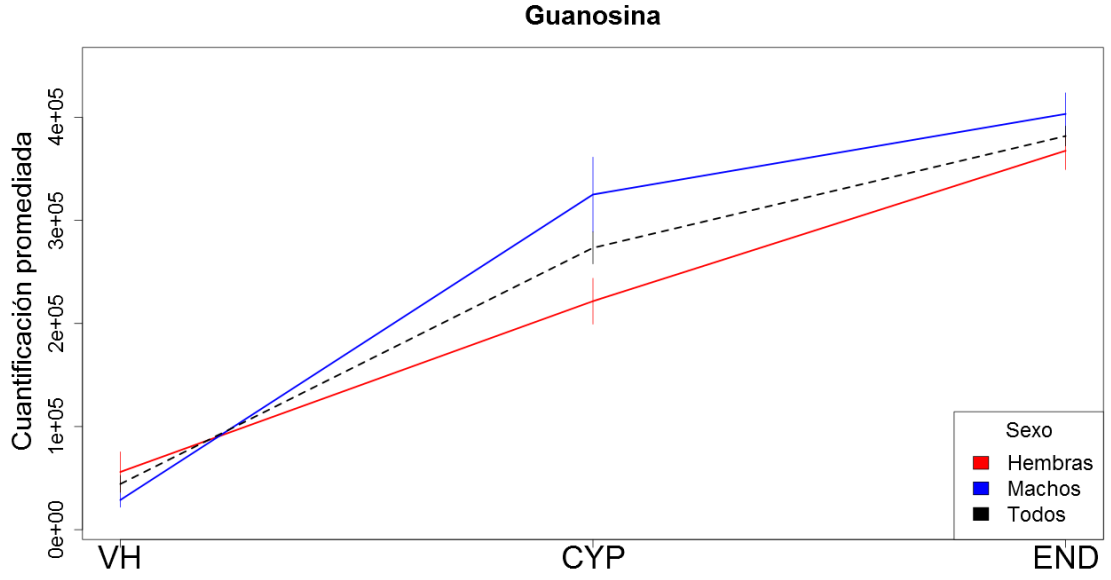


Figura 19. Cuantificación media por tratamiento para la guanosina. Las barras verticales representan el error estándar.

Recurriendo a la bibliografía se encontró un artículo, [58], centrado en el papel de la guanosina en las neuropatologías. Este trabajo indicaba que la guanosina es un compuesto del que se piensa que posee propiedades neuroprotectoras, que se libera en el cerebro en condiciones fisiológicas normales, pero sobre todo durante sucesos patológicos para reducir la neuroinflamación y el estrés oxidativo. Tendría así mucho sentido por ello obtener un aumento significativo de los resultados del test RMRE en los machos ligado con este aumento de la guanosina, que indicaría una posible alteración cerebral como causante de dicho retraso cognitivo.

Respecto a la ornitina, este metabolito presentó una correlación negativa con el RMRE con un valor de -0.8, su comportamiento aparece representado en la Figura 20.

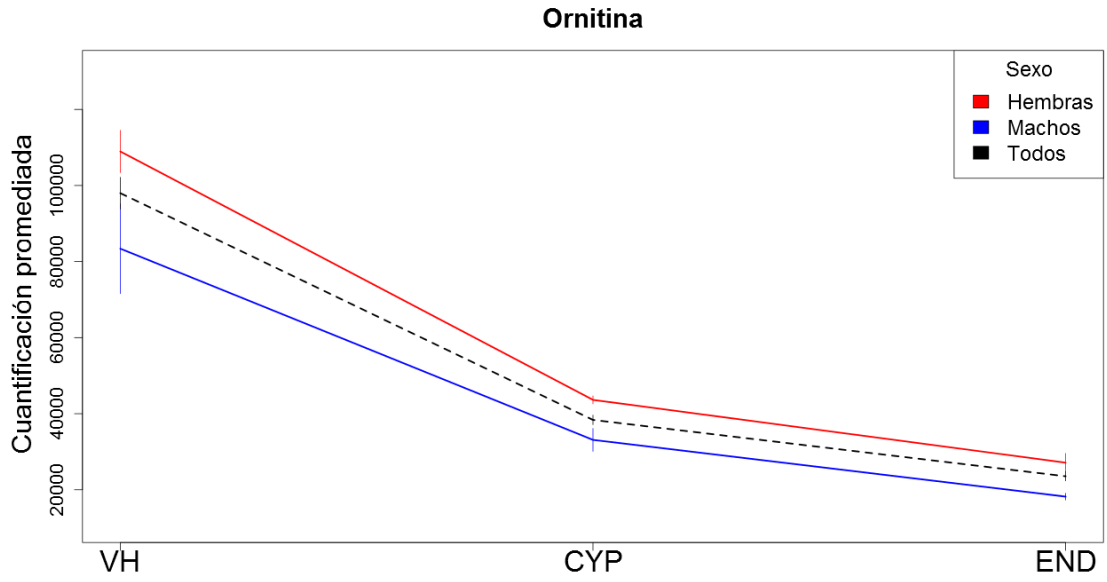


Figura 20. Cuantificación media por tratamiento para la ornitina. Las barras verticales representan el error estándar

La ornitina es uno de los principales compuestos involucrados en el ciclo de la urea, que permite eliminar o controlar los niveles amoniacales en el organismo. Alteraciones en el ciclo de la urea suelen ocasionar un estado de hiperamonemia (HA) [59]. La HA ocasiona cambios en el sistema nervioso central, como son la alteración de las funciones de los neurotransmisores o de los volúmenes de las células del mismo [59] . Es por ello que una disminución de los niveles de ornitina podría derivar en un estado de HA que podría estar dándose en los individuos tratados con Cypermetrín y Endosulfán, y que ello conllevara una alteración neurológica que ocasionara el mayor número de errores de referencia en el test de Radial Maze. No obstante, hay que tener en cuenta que el ciclo de la urea se da en el hígado y se están analizando muestras de cerebelo por lo que tal vez la aproximación no sea certera, aunque pudiera ser que el efecto de la alteración en el hígado fuera sistémico y notificable más allá de él. En otro estudio, [60], se analizó el efecto de la administración in vivo de ornitina en el cerebelo de ratas adolescentes, y se mostraba que niveles elevados de la ornitina pueden afectar a la actividad de una enzima crucial en la neurotransmisión, la Na+, K+-ATPasa. En otro artículo, [61], se estudiaron y demostraron también los efectos de la inyección en el cerebro del isómero L de la ornitina como agente reductor del estrés mental. Estos trabajos motivan a seguir profundizando en el estudio de este metabolito por los posibles efectos que pudieran derivar de su disminución en la mente.

La correlación de la proteína P21707 fue menor que la de los dos metabolitos estudiados, presentando esta un valor de -0.6. En este caso se observa que las ratas tratadas con Cypermetrín eran diferentes a las ratas control y a las tratadas con Endosulfán, siendo estas dos condiciones muy similares entre sí (Figura 21).

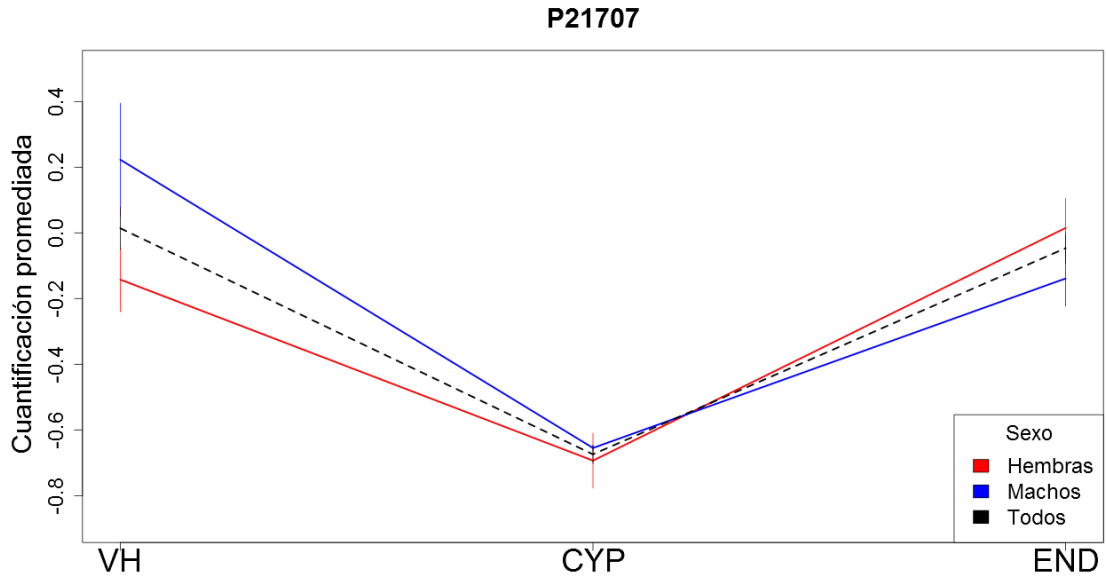


Figura 21. Cuantificación media por tratamiento para la proteína P21707. Las barras verticales representan el error estándar.

Estudiando la información disponible de esta proteína, codificada por el gen Syt1, en Uniprot [62] se observó que está involucrada en una gran cantidad de procesos implicados en el desarrollo y el correcto funcionamiento del cerebro. Participando entre otros en: la regulación del tráfico de vesículas sinápticas en la sinapsis, en la formación de dendritas, en la regulación de concentraciones de calcio (en el análisis de los resultados del Set03 se ha incidido sobre la importancia de este compuesto) además de aparecer explícitamente relacionada con el término GO del desarrollo del cerebro. Por ello, se puede extrapolar que atendiendo a determinadas variables, el efecto del Cypermetrín es relevante frente al del Endosulfán que no muestra ninguna alteración.

Recopilando los resultados del Set01 se podría establecer que los pesticidas Endosulfán y Cypermetrín causan alteraciones cognitivas sobre las ratas macho, y pese a que su efecto pueda ser el mismo sobre determinadas variables se pueden dar los siguientes comportamientos también: que el Cypermetrín altere los niveles de una variable ómica y no lo haga el Endosulfán, o que pese a afectar ambos pesticidas el efecto del Endosulfán sea mayor. Esto demuestra la complejidad de la comprensión de los sistemas biológicos.

5. Discusión

En este TFG se ha presentado una estrategia para el análisis integrativo de datos de distinta índole, en concreto, datos ómicos como proteómica y metabólomica y variables clínicas como los resultados de diversos tests para medir deterioro cognitivo y motor. Este tipo de análisis de integración es uno de los retos actuales de la bioinformática, ya que como se ha constatado en este trabajo, entraña ciertas dificultades.

La primera dificultad encontrada es el pre-procesado de los datos. Los datos ómicos suelen ser bastante ruidosos debido a los procedimientos experimentales y a las técnicas de medición utilizadas. Por ello, es un paso clave del análisis, realizar un pre-tratamiento adecuado que permita sacar el mayor partido posible de los mismos. Un incorrecto diseño del experimento o un pre-procesado inadecuado o insuficiente de los datos podrían ser los causantes de la no obtención de resultados, o lo que podría ser peor, de resultados erróneos. En este trabajo, parte del pre-procesado ya había sido realizado previamente, pero sí que fue necesario aplicar una corrección del ruido y, en ocasiones, una imputación de valores faltantes. También fue útil

realizar un filtrado de las variables ómicas que no presentaban cambios entre las condiciones estudiadas y que, por tanto, iban a tener una contribución nula en el modelo para relacionar los datos ómicos con los resultados de los tests. Aunque, por cuestión de espacio, en muchas ocasiones sólo se han discutido en el trabajo los procedimientos que finalmente se aplicaron, cabe mencionar que se compararon varias estrategias y se seleccionó la que mejores resultados ofrecía a partir de la exploración de datos con el método multivariante PCA. Respecto a los datos de los tests, conviene puntualizar que la variabilidad intra-condiciones es bastante elevada, según se observa en los PCA realizados. Ello puede ser debido a la propia variabilidad biológica en este tipo de medidas o a la falta de precisión de los propios tests, que está acentuada por el reducido tamaño muestral. Por ello, nos centramos en estudiar en más detalle aquellos tests cuyo comportamiento era más robusto al compararlo con resultados previos del laboratorio de Neurobiología obtenidos a partir de muestras mucho más grandes.

La segunda dificultad de los estudios multi-ómicos es la elección de un modelo adecuado que permita incorporar toda la información tanto ómica como de las variables respuesta (los tests en este caso). Escogimos para ello los métodos PLS y *multi-block* PLS en función del número de matrices de datos ómicos disponibles para cada experimento. Este tipo de métodos permiten visualizar de forma efectiva el comportamiento de los datos, gracias a que comportan una reducción de la dimensión de los mismos al seleccionar unas pocas componentes que describen la mayor parte de variabilidad contenida en los datos originales. Además, se han propuesto en este trabajo dos estrategias distintas (MSEP+sPLS y VIP) para seleccionar las variables ómicas con mayor correlación con los resultados de los tests que además, por la propia definición del modelo, presentan los cambios más acusados entre las distintas condiciones experimentales (pesticidas y sexo).

A pesar de las dificultades expuestas y de la baja calidad de algunos de los datos analizados, se ha mostrado la utilidad de los enfoques descritos en el estudio del efecto de los pesticidas en el deterioro de las capacidades neurológicas en crías de ratas expuestas a pesticidas durante el embarazo y lactancia, ya que es posible proponer biomarcadores moleculares que consigan predecir ciertos tipos de alteraciones neurológicas. No obstante, es obvio que sería necesaria una validación experimental de estos candidatos propuestos antes de definirlos como biomarcadores para el uso clínico.

Con respecto a las conclusiones biológicas de los datos analizados, podemos destacar varios aspectos relevantes.

En primer lugar, al comparar las conclusiones obtenidas en cada uno de los dos estudios realizados en este trabajo, llama la atención la gran diferencia entre sexos apreciada en los controles y neutralizadas por la acción de las diferentes concentraciones de Clorpirifos en el Set03, mientras que dicho comportamiento no se observa de forma tan marcada en el Set01. Esto es debido muy probablemente a que el efecto de los pesticidas utilizados en el Set01, Endosulfán y Cypermetrín, es mucho mayor que los estudiados en el Set03 (especialmente el del Endosulfán), por lo que la magnitud de dicho efecto enmascara en parte las diferencias entre sexos que, en cualquier caso, siguen presentes. De hecho, el Endosulfán es el único pesticida de todos los evaluados en este trabajo que tiene prohibido su uso en Estados Unidos y Europa y, tal como hemos observado, es el que tiene mayor efecto en las capacidades cognitivas. Aun así, las diferencias naturales entre machos y hembras, y por ello de sus respuestas cognitivas y motoras, plantea la necesidad de realizar estudios separados por sexo, ya que los resultados en un sexo podrían no ser extrapolables para el otro.

Por otro lado, y atendiendo a que las alteraciones observadas y analizadas afectaban a procesos cognitivos más que a procesos motores, por el estudio de los tests cognitivos de Radial Maze y Morris Water Maze, se podría plantear que tal vez es más probable que los pesticidas ocasionen alteraciones de índole cognitiva que motora, al menos los estudiados en este trabajo. Además, las alteraciones cognitivas siempre se han asociado al hipocampo y no se suelen

encontrar en la literatura trabajos que las relacionen con cambios moleculares en cerebelo. La novedad que también aporta este trabajo es que hemos explorado esta nueva perspectiva, encontrando en ocasiones resultados interesantes.

Así pues, debido a la complejidad, en cuanto a la necesidad de una correcta infraestructura, que entraña la realización de tests para determinar alteraciones en las capacidades cognitivas y motoras, y de las grandes dispersiones en los resultados observadas en muchas ocasiones entre individuos de una misma condición, es fundamental profundizar en el estudio y detección de biomarcadores, tal y como se ha realizado en este trabajo, que permitan detectar de forma rápida, precisa y sencilla dichas alteraciones.

Para conseguir este objetivo, los conocimientos bioinformáticos y estadísticos, junto con los biológicos, son piezas fundamentals para poder manejar, explorar, comprender y validar los resultados de las investigaciones.

6. Conclusiones

Las aportaciones de este TFG se enmarcan en dos ámbitos diferentes: el metodológico y el biológico.

Desde el punto de vista metodológico, se ha puesto a punto una estrategia bioinformática para el análisis multiómico y su integración con otros tipos de variables (resultados de los tests, en este caso) que ha demostrado ser efectiva para extraer información biológica relevante en experimentos de alto rendimiento como son los datos ómicos, que entrarían dentro de la categoría de big data (por el elevado número de variables que se estudian).

Por una parte, se han aplicado distintos procedimientos para el pre-procesado de los datos tales como la reducción del ruido mediante el método ARSyN, la imputación de datos faltantes usando la librería de R *mice*, y el filtrado de proteínas y metabolitos con baja variabilidad entre condiciones usando el coeficiente de variación y el análisis de expresión diferencial mediante la librería de R *limma*.

Por otra parte, se ha propuesto el uso de métodos multivariantes como el PLS y el *multi-block* PLS para obtener modelos que expliquen las alteraciones cognitivas y motoras a partir del comportamiento de las variables ómicas (proteínas y metabolitos). Además, se han diseñado estrategias para extraer las variables ómicas más relevantes en estos modelos a partir, bien de la medida del error de predicción del modelo (MSEP) junto al sPLS, o bien de la medida de la importancia de las variables en la predicción (VIP), que ayudan en la búsqueda de biomarcadores moleculares que puedan predecir los resultados de variables clínicas más difíciles de obtener en algunos casos.

Desde el punto de vista biológico, el análisis de estos experimentos ha permitido entender mejor el efecto de la exposición a pesticidas y del sexo de los sujetos en las alteraciones neurológicas. Se ha constatado que los efectos de Endosulfán, seguidos de Cypermetrin, son más acusados que los de Clorpirifos y que Carbaryl es el pesticida más inocuo, en cuanto a los datos ómicos medidos en cerebelo se refiere. Este comportamiento se ha asociado especialmente a alteraciones cognitivas, donde el deterioro respecto a las ratas control era mayor en machos que en hembras. Así pues, se profundiza en la búsqueda de biomarcadores para deterioro cognitivo y alteraciones neurológicas en machos, proponiendo como candidatos a validar en el laboratorio los siguientes: en cuanto a metabolitos la guanosina y la ornitina; en cuanto a proteínas: P21707, Q64640, D4ADF5 y F1LPH6.

7. Bibliografía

[1] "Proteómica - Medicina molecular." [Online]. Available: http://medmol.es/glosario/75/. [Accessed: 02-Jun-2017].

[2] "What is metabolomics? | EMBL-EBI Train online." [Online]. Available: https://www.ebi.ac.uk/training/online/course/introduction-metabolomics/what-metabolomics. [Accessed: 29-May-2017].

[3] D. Rice, S. Barone, and Jr, "Critical periods of vulnerability for the developing nervous system: evidence from humans and animal models.," *Environ. Health Perspect.*, vol. 108 Suppl 3, no. Suppl 3, pp. 511–33, Jun. 2000.

[4] B. Gómez, "ALTERACIONES NEUROLÓGICAS (COGNITIVAS Y MOTORAS) EN RATAS EXPUESTAS PERINATALMENTE A PESTICIDAS CONTAMINANTES," Centro de Investigación Príncipe Felipe, 2016.

[5] R. M. J. Deacon, "Measuring motor coordination in mice.," *J. Vis. Exp.*, no. 75, p. e2609, May 2013.

[6] R. Ihaka and R. Gentleman, "R: A Language for Data Analysis and Graphics," *J. Comput. Graph. Stat.*, vol. 5, no. 3, pp. 299–314, Sep. 1996.

[7] "CRAN-Home." [Online]. Available: https://cran.r-project.org/. [Accessed: 13-Jun-2017].

[8] "Bioconductor - Home." [Online]. Available: https://www.bioconductor.org/. [Accessed: 13-Jun-2017].

[9] A. R. T. Donders, G. J. M. G. van der Heijden, T. Stijnen, K. G. M. Moons, and H. A. Moll, "Review: a gentle introduction to imputation of missing values.," *J. Clin. Epidemiol.*, vol. 59, no. 10, pp. 1087–91, Oct. 2006.

[10] M. Ringnér, "What is principal component analysis?," *Nat. Biotechnol.*, vol. 26, no. 3, pp. 303–304, Mar. 2008.

[11] Y. Nikolsky, E. Kirillov, R. Zuev, E. Rakhmatulin, and T. Nikolskaya, "Functional Analysis of OMICs Data and Small Molecule Compounds in an Integrated 'Knowledge-Based' Platform," 2009, pp. 177–196.

[12] M. j. Nueda, A. Ferrer, and A. Conesa, "ARSyN: a method for the identification and removal of systematic noise in multifactorial time course microarray experiments," *Biostatistics*, vol. 13, no. 3, pp. 553–566, Jul. 2012.

[13] S. Tarazona *et al.*, "Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package," *Nucleic Acids Res.*, vol. 43, no. 21, p. gkv711, Jul. 2015.

[14] P. Hammer *et al.*, "mRNA-seq with agnostic splice site discovery for nervous system transcriptomics tested in chronic pain," *Genome Res.*, vol. 20, no. 6, pp. 847–860, Jun. 2010.

[15] B. Phipson, S. Lee, I. J. Majewski, W. S. Alexander, and G. K. Smyth, "Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression," *Ann. Appl. Stat.*, vol. 10, no. 2, pp. 946–963, Jun. 2016.

[16] G. Smyth, "Limma Package Introduction," 2004.

[17] R. C. Gentleman *et al.*, "voom: precision weights unlock linear model analysis tools for RNA-seq read counts," *Genome Biol.*, vol. 5, no. 10, p. R80, 2004.

[18] P. Geladi, "Notes on the history and nature of partial least squares (PLS) modelling," *J. Chemom.*, vol. 2, no. January, pp. 231–246, 1988.

[19] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: A basic tool of chemometrics," *Chemom. Intell. Lab. Syst.*, vol. 58, no. 2, pp. 109–130, 2001.

[20] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Anal. Chim. Acta*, vol. 185, no. C, pp. 1–17, 1986.

[21] J. A. Westerhuis, T. Kourti, and J. F. MacGregor, "Analysis of multiblock and hierarchical PCA and PLS models," *J. Chemom.*, vol. 12, no. 5, pp. 301–321, 1998.

[22] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, and P. Besse, "A Sparse PLS for Variable Selection when Integrating Omics Data," *Stat. Appl. Genet. Mol. Biol.*, vol. 7, no. 1, 2008.

[23] H. Chun and S. Keles, "Sparse partial least squares for simultaneous dimension

reduction and variable selection," *J. R. Stat. Soc. Ser. B*, vol. 72, no. 1, pp. 3–25, 2010.
[24] D. M. Allen and D. M. Allen, "American Society for Quality Mean Square Error of Prediction as a Criterion for Selecting Variables American Society for Quality Stable URL : http://www.jstor.org/stable/1267161 Mean Square Error of Prediction as a Criterion for Selecting Variables," vol. 13, no. 3, pp. 469–475, 2016.
[25] N. Akarachantachote, S. Chadcham, K. Saithanu, N. Akarachantachote, S. Chadcham, and K. Saithanu, "CUTOFF THRESHOLD OF VARIABLE IMPORTANCE IN PROJECTION FOR VARIABLE SELECTION," *Int. J. Pure Appl. Math.*, vol. 94, no. 3, pp. 307–322, 2014.
[26] J. Dopazo, "Functional Interpretation of Microarray Experiments," *Omi. A J. Integr. Biol.*, vol. 10, no. 3, pp. 398–410, 2006.
[27] X. Zhou and Z. Su, "EasyGO: Gene Ontology-based annotation and functional enrichment analysis tool for agronomical species," *BMC Genomics*, vol. 8, no. 1, p. 246, 2007.
[28] Gene Ontology Consortium, "The Gene Ontology (GO) database and informatics resource," *Nucleic Acids Res.*, vol. 32, no. 90001, p. 258D–261, Jan. 2004.
[29] "UniProt: a hub for protein information," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D204–D212, Jan. 2015.
[30] E. Bernabeu, "Integration of multi-omics data to describe link between developmental exposure to pesticides and impaired neurodevelopment," Centro de Investigación Príncipe Felipe, 2016.
[31] B. D. Semple, K. Blomgren, K. Gimlin, D. M. Ferriero, and L. J. Noble-Haeusslein, "Brain development in rodents and humans: Identifying benchmarks of maturation and vulnerability to injury across species.," *Prog. Neurobiol.*, vol. 106–107, pp. 1–16, 2013.
[32] A. Peters, "The effects of normal aging on myelin and nerve fibers: a review.," *J. Neurocytol.*, vol. 31, no. 8–9, pp. 581–93.
[33] B. Lakhani *et al.*, "Motor Skill Acquisition Promotes Human Brain Myelin Plasticity.," *Neural Plast.*, vol. 2016, p. 7526135, 2016.
[34] A. Popa-Wagner, S. Mitran, S. Sivanesan, E. Chang, and A.-M. Buga, "ROS and brain diseases: the good, the bad, and the ugly.," *Oxid. Med. Cell. Longev.*, vol. 2013, p. 963520, Dec. 2013.
[35] M. Koga *et al.*, "Glutathione is a physiologic reservoir of neuronal glutamate," *Biochem. Biophys. Res. Commun.*, vol. 409, no. 4, pp. 596–602, Jun. 2011.
[36] C. N. Harada, M. C. Natelson Love, and K. L. Triebel, "Normal cognitive aging.," *Clin. Geriatr. Med.*, vol. 29, no. 4, pp. 737–52, Nov. 2013.
[37] L. S. Deshpande *et al.*, "Alterations in neuronal calcium levels are associated with cognitive deficits after traumatic brain injury.," *Neurosci. Lett.*, vol. 441, no. 1, pp. 115–9, Aug. 2008.
[38] L. M. Veng, M. H. Mesches, and M. D. Browning, "Age-related working memory impairment is correlated with increases in the L-type calcium channel protein α1D (Cav1.3) in area CA1 of the hippocampus and both are ameliorated by chronic nimodipine treatment," *Mol. Brain Res.*, vol. 110, no. 2, pp. 193–202, 2003.
[39] "Tumor Necrosis Factor-α (TNF-α) - Cytokines Growth Factors and Hormones (Obesity) | Sigma-Aldrich." [Online]. Available: http://www.sigmaaldrich.com/life-science/cell-biology/cell-biology-products.html?TablePage=14576697. [Accessed: 07-Jun-2017].
[40] Y. Dowlati *et al.*, "A Meta-Analysis of Cytokines in Major Depression," *Biol. Psychiatry*, vol. 67, no. 5, pp. 446–457, Mar. 2010.
[41] M. J. Stuart and B. T. Baune, "Chemokines and chemokine receptors in mood disorders, schizophrenia, and cognitive impairment: A systematic review of biomarker studies," *Neurosci. Biobehav. Rev.*, vol. 42, pp. 93–115, May 2014.
[42] V. M.-Y. Lee, M. Goedert, and J. Q. Trojanowski, "Neurodegenerative Tauopathies," *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 1121–1159, Mar. 2001.
[43] K. Iqbal, F. Liu, C.-X. Gong, A. del C. Alonso, and I. Grundke-Iqbal, "Mechanisms of tau-induced neurodegeneration," *Acta Neuropathol.*, vol. 118, no. 1, pp. 53–69, Jul. 2009.

[44] A. C. McKee *et al.*, “Chronic Traumatic Encephalopathy in Athletes: Progressive Tauopathy After Repetitive Head Injury,” *J. Neuropathol. Exp. Neurol.*, vol. 68, no. 7, pp. 709–735, Jul. 2009.
[45] C. Li, S. Liu, Y. Xing, and F. Tao, “The role of hippocampal tau protein phosphorylation in isoflurane-induced cognitive dysfunction in transgenic APP695 mice.,” *Anesth. Analg.*, vol. 119, no. 2, pp. 413–9, Aug. 2014.
[46] M. E. Murray *et al.*, “Clinicopathologic and [11] C-Pittsburgh compound B implications of Thal amyloid phase across the Alzheimer’s disease spectrum,” *Brain*, vol. 138, no. 5, pp. 1370–1381, May 2015.
[47] A. d. C. Alonso, T. Zaidi, M. Novak, I. Grundke-Iqbal, and K. Iqbal, “Hyperphosphorylation induces self-assembly of into tangles of paired helical filaments/straight filaments,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 12, pp. 6923–6928, Jun. 2001.
[48] M. Picard and B. S. McEwen, “Mitochondria impact brain function and cognition.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 1, pp. 7–8, Jan. 2014.
[49] D. C. Wallace, “Bioenergetics, the origins of complexity, and the ascent of man,” *Proc. Natl. Acad. Sci.*, vol. 107, no. Supplement_2, pp. 8947–8953, May 2010.
[50] M. Picard and D. M. Turnbull, “Linking the Metabolic State and Mitochondrial DNA in Chronic Disease, Health, and Aging,” *Diabetes*, vol. 62, no. 3, pp. 672–678, Mar. 2013.
[51] F. Gomez-Pinilla and C. Hillman, “The Influence of Exercise on Cognitive Abilities,” *Compr. Physiol.*, vol. 3, no. 1, p. 403, 2013.
[52] A. Citri and R. C. Malenka, “Synaptic Plasticity: Multiple Forms, Functions, and Mechanisms,” *Neuropsychopharmacology*, vol. 33, no. 1, pp. 18–41, Jan. 2008.
[53] H. Hori *et al.*, “Cognitive effects of the ANK3 risk variants in patients with bipolar disorder and healthy individuals,” *J. Affect. Disord.*, vol. 158, pp. 90–96, Apr. 2014.
[54] S. Rangaraju *et al.*, “Mood, stress and longevity: convergence on ANK3,” *Mol. Psychiatry*, vol. 21, no. 8, pp. 1037–1049, Aug. 2016.
[55] V. Vijay, T. Han, C. L. Moland, J. C. Kwekel, J. C. Fuscoe, and V. G. Desai, “Sexual dimorphism in the expression of mitochondria-related genes in rat heart at different ages.,” *PLoS One*, vol. 10, no. 1, p. e0117047, 2015.
[56] S. M. Mense *et al.*, “The Common Insecticides Cyfluthrin and Chlorpyrifos Alter the Expression of a Subset of Genes with Diverse Functions in Primary Human Astrocytes,” *Toxicol. Sci.*, vol. 93, no. 1, pp. 125–135, Jun. 2006.
[57] M. Santos-Galindo, E. Acaz-Fonseca, M. J. Bellini, and L. M. Garcia-Segura, “Sex differences in the inflammatory response of primary astrocytes to lipopolysaccharide,” *Biol. Sex Differ.*, vol. 2, no. 1, p. 7, Jul. 2011.
[58] L. E. B. Bettio, J. Gil-Mohapel, and A. L. S. Rodrigues, “Guanosine and its role in neuropathologies.,” *Purinergic Signal.*, vol. 12, no. 3, pp. 411–26, Sep. 2016.
[59] A. L. Gropman, M. Prust, A. Breeden, S. Fricke, and J. VanMeter, “Urea cycle defects and hyperammonemia: effects on functional imaging.,” *Metab. Brain Dis.*, vol. 28, no. 2, pp. 269–75, Jun. 2013.
[60] ?ngela Zanatta *et al.*, “Ornithine In Vivo Administration Disrupts Redox Homeostasis and Decreases Synaptic Na+, K+-ATPase Activity in Cerebellum of Adolescent Rats: Implications for the Pathogenesis of Hyperornithinemia-Hyperammonemia-Homocitrullinuria (HHH) Syndrome,” *Cell. Mol. Neurobiol.*, vol. 35, no. 6, pp. 797–806, Aug. 2015.
[61] R. Suenaga *et al.*, “Central l-arginine reduced stress responses are mediated by l-ornithine in neonatal chicks,” *Amino Acids*, vol. 35, no. 1, pp. 107–113, Jun. 2008.
[62] “UniProtKB-P21707(SYT1_RAT).” [Online]. Available: http://www.uniprot.org/uniprot/P21707. [Accessed: 14-Jun-2017].

8. Adjuntos: Ejemplos códigos

A continuación se adjuntan ejemplos de los códigos más relevantes que se elaboraron y emplearon durante el TFG para el análisis de datos y la obtención de resultados.

8.1 Imputación de VFs

```
#############################################################
##### Inputacion datos SET03  por mice ####
#############################################################

library(mice)

# Imputacion por MICE ---------------------------------------------
------
#CB
prot03_CB <-
  read.delim(

"~/Dropbox/denamic/data/set03/00_Set03_CB_ListaParaInputacionNAs.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

sum(apply(prot03_CB, 2, function(colum) {
  sum(is.na(colum))
}))##2964

prot03_CB_Imputada <- mice(t(prot03_CB), method = "norm.predict")
# #despues deberia hacer: complete(prot03_CB_Imputada) para obtener la
matriz imputada

prot03_CB_Imputada <- complete(prot03_CB_Imputada)
sum(apply(prot03_CB_Imputada, 2, function(colum) {
  sum(is.na(colum))
}))##0

prot03_CB_Imputada <- t(prot03_CB_Imputada)

#Guardar la matriz para PCA sin normalizar
write.table(
  prot03_CB_Imputada,
  file =
"~/Dropbox/denamic/data/set03/01_Set03_CB_Imputada_Por_MICE.txt",
  col.names = T,
  row.names = T
)

#HP
prot03_HP <-
  read.delim(

"~/Dropbox/denamic/data/set03/00_Set03_HP_ListaParaInputacionNAs.txt",
    header = T,
    row.names = 1,
    sep = " "
  )
```

```r
prot03_HP_Imputada <- mice(t(prot03_HP), method = "norm.predict")

prot03_HP_Imputada <- complete(prot03_HP_Imputada)
sum(apply(prot03_HP_Imputada, 2, function(colum) {
  sum(is.na(colum))
}))##0

prot03_HP_Imputada <- t(prot03_HP_Imputada)

#Guardar la matriz para PCA sin normalizar
write.table(
  prot03_HP_Imputada,
  file =
"~/Dropbox/denamic/data/set03/01_Set03_HP_Imputada_Por_MICE.txt",
  col.names = T,
  row.names = T
)
```

8.2 PCAs

```
############################
## PCAs SET01 #############
############################

#to get the colors linked to pestides and shapes linked to gender
source("~/Dropbox/denamic/scripts/Pesticides_Colors_and_GenderShapes.R
")
library(mixOmics) #to obtain the PCA'S


#################PROTEOMICA##############################

#SET01
#CB
prot01_CB <-
  read.delim("~/Dropbox/denamic/data/set01/00_Set01_prot_CB.txt")

#homogenizacion de nombres
noms <- colnames(prot01_CB)
noms <- gsub(".", "_", noms, fixed = T)
colnames(prot01_CB) <- gsub(".", "_", colnames(prot01_CB), fixed = T)
prot01_CB <- t(prot01_CB)
par(mar = c(8, 2, 4, 2))
boxplot(t(prot01_CB),
        las = 2,
        main = "Set 01 - Proteomics- CB",
        col = "grey")

# Design matrix ------------------------------------------------------
-----

##MATRIZ DE DISEÑO
ids <- rownames(prot01_CB)
designProt01_CB <- do.call("rbind", strsplit(ids, "_", fixed = T))
rownames(designProt01_CB) <- ids
colnames(designProt01_CB) <-
  c("Pesticide", "Gender", "Replicate", "Tissue")
ncol(prot01_CB)


# Saving design matrix -----------------------------------------------
-----
write.table(
  designProt01_CB,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designProt01_CB.txt",
  row.names = T,
  col.names = T
)

X <- prot01_CB
boxplot(X)
boxplot(scale(X, center = TRUE, scale = TRUE))

# PCAs and loading plots scaled --------------------------------------
-----

##con Scale=T
pcaX <- pca(X,
```

```
              ncomp = 3,
              center = TRUE,
              scale = TRUE)

par(mar = c(5, 4, 2, 2))

####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingprot01_cb_scaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingprot01_cb_scaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAprot01_cb_scaled")
plotIndiv(
  pcaX,
  group = designProt01_CB[, 1],
  ind.names = FALSE,
  title = "Set 01 - Proteomics - CB-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designProt01_CB[, 2]],
  col.per.group = MYcolors[designProt01_CB[, 1]]
)

#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designProt01_CB[, 1]),
       col = unique(MYcolors[designProt01_CB[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designProt01_CB[, 2]),
  pch = unique(myPCH[designProt01_CB[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()

## to check if the simbols and shapes are well assigned
plotIndiv(
  pcaX,
  group = factor(designProt01_CB[, 1], levels = c("VH", "END",
"CYP")),
  ind.names = rownames(designProt01_CB),
  legend = F,
  title = "Set01_Prot_CB_withIDS",
  ellipse = F,
  col.per.group = MYcolors[c("VH", "END", "CYP")]
)


# PCAs and loading plots NON scaled ---------------------------------
-----
```

```r
#con Scale=F

pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = F)

par(mar = c(5, 4, 2, 2))


####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingprot01_cb_NONscaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingprot01_cb_NONscaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAprot01_cb_NONscaled")
plotIndiv(
  pcaX,
  group = factor(designProt01_CB[, 1], levels = c("VH", "END",
"CYP")),
  ind.names = FALSE,
  title = "Set 01 - Proteomics - CB-NonScaled",
  size.title = rel(1),
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designProt01_CB[, 2]],
  col.per.group = MYcolors[c("VH", "END", "CYP")]
)

#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designProt01_CB[, 1]),
       col = unique(MYcolors[designProt01_CB[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designProt01_CB[, 2]),
  pch = unique(myPCH[designProt01_CB[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()


# HP ----------------------------------------------------------------
-----
prot01_HP <-
  read.delim("~/Dropbox/denamic/data/set01/00_Set01_prot_HP.txt")

#homogenizacion de nombres
noms <- colnames(prot01_HP)
noms <- gsub(".", "_", noms, fixed = T)
colnames(prot01_HP) <- gsub(".", "_", colnames(prot01_HP), fixed = T)
prot01_HP <- t(prot01_HP)
```

```
par(mar = c(8, 2, 4, 2))
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Prot01_HP")
boxplot(t(prot01_HP),
        las = 2,
        main = "Set 01 - Proteomics- HP",
        col = "grey")
dev.off()

# Design matrix ------------------------------------------------------
-----
ids <- rownames(prot01_HP)
designProt01_HP <- do.call("rbind", strsplit(ids, "_", fixed = T))
rownames(designProt01_HP) <- ids
colnames(designProt01_HP) <-
  c("Pesticide", "Gender", "Replicate", "Tissue")

# Saving design matrix -----------------------------------------------
-----
write.table(
  designProt01_HP,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designProt01_HP.txt",
  row.names = T,
  col.names = T
)

X <- prot01_HP
boxplot(X)
boxplot(scale(X, center = TRUE, scale = TRUE))

# PCAs and loading plots scaled --------------------------------------
-----

##con Scale=T
pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = TRUE)

par(mar = c(5, 4, 2, 2))

####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingprot01_HP_scaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingprot01_HP_scaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAprot01_HP_scaled")
plotIndiv(
  pcaX,
  group = designProt01_HP[, 1],
  ind.names = FALSE,
  title = "Set 01 - Proteomics - HP-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designProt01_HP[, 2]],
```

```
  col.per.group = MYcolors[designProt01_HP[, 1]]
)


#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designProt01_HP[, 1]),
       col = unique(MYcolors[designProt01_HP[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designProt01_HP[, 2]),
  pch = unique(myPCH[designProt01_HP[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()

# PCAs and loading plots NON scaled ---------------------------------
-----

#con Scale=F

pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = F)

par(mar = c(5, 4, 2, 2))


####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingprot01_HP_NONscaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingprot01_HP_NONscaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAprot01_HP_NONscaled")
plotIndiv(
  pcaX,
  group = designProt01_HP[, 1],
  ind.names = FALSE,
  title = "Set 01 - Proteomics -HP-NonScaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designProt01_HP[, 2]],
  col.per.group = MYcolors[designProt01_HP[, 1]]
)

#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designProt01_HP[, 1]),
       col = unique(MYcolors[designProt01_HP[, 1]]),
       pch = myPCH)
```

```r
#to plot the shape according to sex
legend(
  "topleft",
  unique(designProt01_HP[, 2]),
  pch = unique(myPCH[designProt01_HP[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()

# Metabolomics ------------------------------------------------------
-----

# CB ----------------------------------------------------------------
-----
met01_CB <-
  read.delim("~/Dropbox/denamic/data/set01/00_Set01_met_CB.txt")

#homogenizacion de nombres
noms <- colnames(met01_CB)
noms <- gsub(".", "_", noms, fixed = T)
colnames(met01_CB) <- gsub(".", "_", colnames(met01_CB), fixed = T)
met01_CB <- t(met01_CB)
par(mar = c(8, 2, 4, 2))
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Met01_CB")
boxplot(
  t(met01_CB),
  las = 2,
  main = "Set 01 -Metabolomics- CB",
  col = "grey",
  cex.axis = 0.7
)
dev.off()


# Design matrix -----------------------------------------------------
-----
ids <- rownames(met01_CB)
designMet01_CB <- do.call("rbind", strsplit(ids, "_", fixed = T))

rownames(designMet01_CB) <- ids
colnames(designMet01_CB) <-
  c("Pesticide", "Gender", "Replicate", "Tissue")


# Saving design matrix ----------------------------------------------
-----
write.table(
  designMet01_CB,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designMet01_CB.txt",
  row.names = T,
  col.names = T
)

##boxplot to see if it is necessary scaling
X <- met01_CB
```

```
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Met01_CB_IsScallingNec
essary")
boxplot(
  X,
  xlab = "Metabolites",
  main = "Set 01 -Scalling?-Metabolomics- CB",
  col = "grey",
  cex.axis = 0.7
)
dev.off()
boxplot(scale(X, center = TRUE, scale = F))

# PCAs and loading plots scaled ----------------------------------------
-----

##con Scale=T
pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = TRUE)

par(mar = c(5, 4, 2, 2))

####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingMet01_cb_scaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingMet01_cb_scaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAMet01_cb_scaled")
plotIndiv(
  pcaX,
  group = designMet01_CB[, 1],
  ind.names = FALSE,
  title = "Set 01 - Metabolomics - CB-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designMet01_CB[, 2]],
  col.per.group = MYcolors[designMet01_CB[, 1]]
)

#to plot the  colour according to pesticide
# legend("bottomleft", c("VH", "PEST"), col = c(1,2), pch = myPCH)
legend("bottomleft",
       unique(designMet01_CB[, 1]),
       col = unique(MYcolors[designMet01_CB[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designMet01_CB[, 2]),
  pch = unique(myPCH[designMet01_CB[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)
```

```
dev.off()

## to check if the simbols and shapes are well assigned
plotIndiv(
  pcaX,
  group = designMet01_CB[, 1],
  ind.names = rownames(designMet01_CB),
  legend = F,
  title = "Set01_Met_CB_withIDS",
  ellipse = F,
  col.per.group = MYcolors[designMet01_CB[, 1]]
)


# PCAs and loading plots NON scaled ----------------------------------
-----

#con Scale=F

pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = F)

par(mar = c(5, 4, 2, 2))

####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingMet01_cb_NONscaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingMet01_cb_NONscaled")
dev.off()

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAMet01_cb_NONscaled")
plotIndiv(
  pcaX,
  group = designMet01_CB[, 1],
  ind.names = FALSE,
  title = "Set 01 - Metabolomics - CB-NonScaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designMet01_CB[, 2]],
  col.per.group = MYcolors[designMet01_CB[, 1]]
)

#to plot the  colour according to pesticide
# legend("bottomleft", c("VH", "PEST"), col = c(1,2), pch = myPCH)
legend("bottomleft",
       unique(designMet01_CB[, 1]),
       col = unique(MYcolors[designMet01_CB[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designMet01_CB[, 2]),
  pch = unique(myPCH[designMet01_CB[, 2]]) ,
  bty = "o",
```

```
  ncol = 2,
  col = "gray45"
)

dev.off()

# HP ----------------------------------------------------------------
-----
met01_HP <-
  read.delim("~/Dropbox/denamic/data/set01/00_Set01_met_HP.txt")
View(met01_HP)

#homogenizacion de nombres
noms <- colnames(met01_HP)
noms <- gsub(".", "_", noms, fixed = T)
colnames(met01_HP) <- gsub(".", "_", colnames(met01_HP), fixed = T)
met01_HP <- t(met01_HP)
par(mar = c(8, 2, 4, 2))
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Met01_HP")
boxplot(
  t(met01_HP),
  las = 2,
  main = "Set 01 -Metabolomics- HP",
  col = "grey",
  cex.axis = 0.7
)
dev.off()

# Design matrix -----------------------------------------------------
-----
ids <- rownames(met01_HP)
designMet01_HP <- do.call("rbind", strsplit(ids, "_", fixed = T))
rownames(designMet01_HP) <- ids
colnames(designMet01_HP) <-
  c("Pesticide", "Gender", "Replicate", "Tissue")

# Saving design matrix ----------------------------------------------
-----
write.table(
  designMet01_HP,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designMet01_HP.txt",
  row.names = T,
  col.names = T
)

##boxplot to see if it is necessary scaling
X <- met01_HP
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Met01_HP_IsScallingNec
essary")
boxplot(
  X,
  xlab = "Metabolites",
  main = "Set 01 -Scalling?-Metabolomics- HP",
  col = "grey",
  cex.axis = 0.7
)
dev.off()
boxplot(scale(X, center = TRUE, scale = F))
```

```r
# PCAs and loading plots scaled ----------------------------------------
-----

##con Scale=T

pcaX <-
  pca(X,
      ncomp = 3,
      center = TRUE,
      scale = TRUE)##Error: cannot rescale a constant/zero column to
unit variance.

###REMOVING COLUMNS WITH NO VARIANCE
#calculo de las varianzas para cada metabolito
VarianceMetabolites <- apply(X, 2, var)
#identificacion de los metabolitos cuya varianza==0
a <- 0
MetabolitesWithNoVariance <- character()
for (element in VarianceMetabolites) {
  a <- a + 1
  if (element == 0) {
    cat(element, " ", names(VarianceMetabolites[a]) , "\n")
    MetabolitesWithNoVariance <-
      c(MetabolitesWithNoVariance, names(VarianceMetabolites[a]))
  }
}
#indices de las columnas correspondientes a los metabolitos cuya
varianza=0
indexes <- NULL
for (element in MetabolitesWithNoVariance) {
  indexes <- c(indexes, (which(colnames(X) == element)))
}
####las columnas sin varianza son la 16 y la 19, se corresponden con
"5-HIAA" y "ACETYLCHOLINE"
#eliminacion de las columnas cuya varianza=0
X <- X[, -indexes]

# Saving Metabolomic HP data with columns with no Variance deleted ---
-----
write.table(X ,
            file = "~/Dropbox/denamic/data/set01/01_Set01_met_HP.txt",
            row.names = T,
            col.names = T)

#CONTINUAMOS CON EL PCA
pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = TRUE)

###hacer loadings tb para ver donde tienden a acumularse, queremos que
la mayoria esten sobre el 0
par(mar = c(5, 4, 2, 2))

png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingMet01_HP_scaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingMet01_HP_scaled")
dev.off()
```

```r
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/PCAMet01_HP_scaled")
plotIndiv(
  pcaX,
  group = designMet01_HP[, 1],
  ind.names = FALSE,
  title = "Set 01 - Metabolomics -HP-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designMet01_HP[, 2]],
  col.per.group = MYcolors[designMet01_HP[, 1]]
)


#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designMet01_HP[, 1]),
       col = unique(MYcolors[designMet01_HP[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designMet01_HP[, 2]),
  pch = unique(myPCH[designMet01_HP[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()

# Clinical data analysis ---------------------------------------------
-----
clinic01 <-
  read.delim("~/Dropbox/denamic/data/set01/00_Set01_clinic.txt", sep =
" ")
a <- clinic01
View(clinic01)
rownames(clinic) == rownames(designClinic01)
rownames(clinic)[21]
sum(rownames(designClinic01) == "END_M_III5")
clinic <- clinic[-21, ]
clinic01 <- clinic
X <- clinic01
write.table(clinic01,
            file = "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
            col.names = T,
            row.names = T)

X <-
  read.delim(
    "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )
```

```
# Design matrix ------------------------------------------------------
-----

##MATRIZ DE DISEÑO
ids <- rownames(clinic01)
designClinic01 <- do.call("rbind", strsplit(ids, "_", fixed = T))
rownames(designClinic01) <- ids
colnames(designClinic01) <- c("Pesticide", "Gender", "Replicate")

# Saving design matrix -----------------------------------------------
-----
write.table(
  designClinic01,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designClinic01.txt",
  row.names = T,
  col.names = T
)

X <-
  read.delim(
    "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

designClinic01 <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designClinic01.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

##boxplot to see if it is necessary scaling
X <- t(clinic01)
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Clinic01_IsScallingNec
essary")
boxplot(
  X,
  main = "Set 01 -Scalling?-Clinic",
  col = "grey",
  cex.axis = 0.7,
  las = 2
)

which(colnames(X) == "END_M_III5")
X <- X[, -13]

dev.off()
boxplot(scale(X, center = TRUE, scale = F))

Xscaled = scale(X, center = TRUE, scale = TRUE)

boxplot(scale(X, center = TRUE, scale = TRUE))

#es necesario escalar
X <-
```

```
  read.delim(
    "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

designClinic01 <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designClinic01.txt",
    header = T,
    row.names = 1,
    sep = " "
  )


# PCAs and loading plots scaled ----------------------------------------
-----

##con Scale=T
pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = TRUE)

par(mar = c(5, 4, 2, 2))

####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingClinic01_scaled")
plot(pcaX$loadings$X[, 1:2], main = "loadingClinic01_scaled")
dev.off()

png(filename = "~/Dropbox/denamic/plots/Set01/Pcas/Clinic01_scaled")
plotIndiv(
  pcaX,
  group = designClinic01[, 1],
  ind.names = FALSE,
  title = "Set 01 -Clinic01-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designClinic01[, 2]],
  col.per.group = MYcolors[designClinic01[, 1]]
)

#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designClinic01[, 1]),
       col = unique(MYcolors[designClinic01[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designClinic01[, 2]),
  pch = unique(myPCH[designClinic01[, 2]]) ,
  bty = "o",
  ncol = 2,
```

```
  col = "gray45"
)

dev.off()

##I'M GOING TO DELETE THE "OUTLIERS" VALUE TO PLOT AGAIN THE PCA
x11()
plotIndiv(
  pcaX,
  group = designClinic01[, 1],
  ind.names = rownames(designClinic01),
  legend = F,
  title = "Set01_Clinic01_Scaled_WithIDS",
  ellipse = F,
  col.per.group = MYcolors[designClinic01[, 1]]
)

##OUTLIERS DETECTED: ("END_M_III5", "VH_F_I5"),vamos a eliminarlos y
comenzar de nuevo el PCA Analisis
outliers <- c("END_M_III5", "VH_F_I5")
indexes <- NULL
for (element in outliers) {
  indexes <- c(indexes, which(rownames(X) == element))
}

clinic01 <- clinic01[-indexes, ]
# Design matrix ------------------------------------------------------
-----

##MATRIZ DE DISEÑO
ids <- rownames(clinic01)
designClinic01 <- do.call("rbind", strsplit(ids, "_", fixed = T))
rownames(designClinic01) <- ids
colnames(designClinic01) <- c("Pesticide", "Gender", "Replicate")


# Saving design matrix -----------------------------------------------
-----
write.table(
  designClinic01,
  file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designClinic01.txt",
  row.names = T,
  col.names = T
)

##boxplot to see if it is necessary scaling
X <- clinic01
png(filename =
"~/Dropbox/denamic/plots/Set01/Boxplots/Boxplot_Clinic01_IsScallingNec
essary")
boxplot(
  X,
  main = "Set 01 -Scalling?-Clinic",
  col = "grey",
  cex.axis = 0.7,
  las = 2
)

dev.off()
boxplot(scale(X, center = TRUE, scale = F))
```

```
Xscaled = scale(X, center = TRUE, scale = TRUE)

boxplot(scale(X, center = TRUE, scale = TRUE))

#es necesario escalar

# PCAs and loading plots scaled ----------------------------------------
-----
X <-
  read.delim(
    "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

designClinic01 <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set01/DesignMatrixes/designClinic01.txt",
    header = T,
    row.names = 1,
    sep = " "
  )
gsub("...", "_", colnames(X))
colnames(X)

X <-
  read.delim(
    "~/Dropbox/denamic/data/set01/01_Set01_clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )

##con Scale=T
pcaX <- pca(X,
            ncomp = 3,
            center = TRUE,
            scale = TRUE)

par(mar = c(5, 4, 2, 2))


####hacer loadings tb para ver donde tienden a acumularse, queremos
que la mayoria esten sobre el 0
png(filename =
"~/Dropbox/denamic/plots/Set01/Pcas/loadingClinic01_scaled")

plot(
  pcaX$loadings$X[, 1:2],
  main = "loadingClinic01_scaled",
  xlim = c(0.2, 0.9),
  col = "red"
)
text(
  pcaX$loadings$X[, 1:2][, 1],
  y = pcaX$loadings$X[, 1:2][, 2] ,
  labels = rownames(pcaX$loadings$X),
  pos = 4
```

```
)
dev.off()

png(filename = "~/Dropbox/denamic/plots/Set01/Pcas/Clinic01_scaled")
plotIndiv(
  pcaX,
  group = designClinic01[, 1],
  ind.names = FALSE,
  title = "Set 01 -Clinic01-Scaled",
  size.title = 1.2,
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designClinic01[, 2]],
  col.per.group = MYcolors[designClinic01[, 1]]
)

#to plot the  colour according to pesticide
legend("bottomleft",
       unique(designClinic01[, 1]),
       col = unique(MYcolors[designClinic01[, 1]]),
       pch = myPCH)
#to plot the shape according to sex
legend(
  "topleft",
  unique(designClinic01[, 2]),
  pch = unique(myPCH[designClinic01[, 2]]) ,
  bty = "o",
  ncol = 2,
  col = "gray45"
)

dev.off()
```

8.3 Correción de ruido, ARSyN

```
#########################################################################
#######
################          ARSYNSEQ
################################
#########################################################################
#######
library(NOISeq)

#####ADDING TO THE DESIGN MATRIXES THE PEST_GENDER COLUMN ------------
---

#CB
design_CB <-
  read.delim(
    "~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_CB",
    header = T,
    row.names = 1,
    sep = " "
  )
data <- design_CB
data$Pest_Gender <-
  apply(data[, c(1, 2)] , 1 , paste , collapse = "_")

###Saving Design Matrix for Arsyn CB
write.table(
  data,

"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_CB.txt
",
  col.names = T,
  row.names = T,
  sep = " ",
  quote = F
)

#HP
design_HP <-
  read.delim(
    "~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_HP",
    header = T,
    row.names = 1,
    sep = " "
  )
data <- design_HP
data$Pest_Gender <-
  apply(data[, c(1, 2)] , 1 , paste , collapse = "_")

###Saving Design Matrix for Arsyn HP
write.table(
  data,

"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_HP.txt
",
  col.names = T,
  row.names = T,
  sep = " ",
  quote = F
)
```

```
# applying ARSYNSEQ  ------------------------------------------------
-----

# #CB ----------------------------------------------------------------
-----
protCB <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set03/01_Set03_CB_Imputada_Por_MICE.txt",
    sep = " ",
    header = T,
    row.names = 1
  )
protCB <- t(protCB)

designCB <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_CB.txt
",
    sep = " ",
    header = T,
    row.names = 1
  )
design <- designCB

prot <- readData(data = protCB, factors = design)

prot_postARSYN <-
  ARSyNseq(
    data = prot,
    factor = "Pest_Gender",
    logtransf = TRUE,
    norm = "n"
  )
prot_postARSYNmatrix <- prot_postARSYN@assayData$exprs

prot_postARSYNmatrix <- t(prot_postARSYNmatrix)

###Saving Design Matrix for Arsyn CB
write.table(
  prot_postARSYNmatrix,
  "~/Dropbox/denamic/data/set03/02_Set03_CB_PostArsyn",
  col.names = T,
  row.names = T,
  sep = " ",
  quote = F
)

# #HP ----------------------------------------------------------------
-----
protHP <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set03/01_Set03_HP_Imputada_Por_MICE.txt",
    sep = " ",
    header = T,
    row.names = 1
  )
protHP <- t(protHP)
```

```
designHP <-
  read.delim(
    file =
"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03 ARSYN HP.txt
",
    sep = " ",
    header = T,
    row.names = 1
  )
design <- designHP

prot <- readData(data = protHP, factors = design)

prot_postARSYN <-
  ARSyNseq(
    data = prot,
    factor = "Pest_Gender",
    logtransf = TRUE,
    norm = "n"
  )
prot_postARSYNmatrix <- prot_postARSYN@assayData$exprs

prot_postARSYNmatrix <- t(prot_postARSYNmatrix)

###Saving Design Matrix for Arsyn CB
write.table(
  prot_postARSYNmatrix,
  "~/Dropbox/denamic/data/set03/02_Set03_HP_PostArsyn",
  col.names = T,
  row.names = T,
  sep = " ",
  quote = F
)
```

8.4 PLS

```
########################################
### Estrategia Seleccion Variables ########
########################################

library(mixOmics)
library(mice)

source("~/Dropbox/denamic/scripts/Pesticides_Colors_and_GenderShapes.R
")

# CB ----------------------------------------------------------------
-----

#lectura matriz X=proteomics
cb <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_CB_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
cb <- cb[-which(rownames(cb) == "CYPER_M_I6_CB"), ]
rownames(cb) <- gsub("_CB", "", rownames(cb))

#lectura matriz Y=clinical
clinical <-
  read.delim(
    "~/Dropbox/denamic/data/set03/00_Set03_Clinic.txt",
    header = T,
    row.names = 1,
    sep = " "
  )
clinical <- clinical[-which(rownames(clinical) == "CYP_M_I6"), ]

#que observaciones son distintas entre cada matriz
#LOS    que estan en clinical y NO en cb,#2 observaciones
distintas:""VH_M_I4"" "CHLOR01_M_mM"
noestan <- NULL
for (element in rownames(clinical)) {
  if (sum(rownames(cb) == element) == 0) {
    print(element)
    noestan <- c(element, noestan)
  }
}

posNoestan <- NULL

for (element in noestan) {
  posNoestan <- c(which(rownames(clinical) == element), posNoestan)
}
posNoestan#6,23
clinical <- clinical[-posNoestan, ]

#los que estan en cb Y NO en clinical :"CHLOR01_M_I1"
noestan <- NULL
for (element in rownames(cb)) {
  if (sum(rownames(clinical) == element) == 0) {
    print(element)
    noestan <- c(element, noestan)
```

```
  }
}
posNoestan <- NULL
for (element in noestan) {
  posNoestan <- c(which(rownames(cb) == element), posNoestan)
}
posNoestan#6
cb <- cb[-posNoestan, ]

#ORDENADO FILAS MATRIZ CLINICAL, SEGUN EL ORDEN DE CB
clinical <- clinical[rownames(cb), , drop = FALSE]

###eliminando Radial Maze de clinical
clinical <- clinical[, 1:3]

###Imputacion valores faltantes de clinical por MICE
sum(apply(clinical, 2, function(colum) {
  sum(is.na(colum))
}))##14

imputed.clinical <- mice(clinical, method = "norm.predict")
# #despues deberia hacer: complete(prot03_CB_Imputada) para obtener la
matriz imputada
imputed.clinical <- complete(imputed.clinical)

clinical.org <- clinical

clinical <- imputed.clinical

#guardado matriz imputada de clinical
write.table(
  clinical,
  file =
"~/Dropbox/denamic/data/set03/02_Set03_ClinicalImputedByMice.txt",
  col.names = T,
  row.names = T,
  sep = "\t"
)

# centrado y escalado matrices ----------------------------------------
-----
#PROTEOM
cb <- scale(cb, center = TRUE, scale = FALSE)
#CLINICAL DATA
clinical <- scale(clinical, center = TRUE, scale = TRUE)

####CB QUEDARME SOLO CON LOS GENES CON VARIABILIDAD ELEVADA:TOTAL
TOTAL
load(

"~/Dropbox/denamic/data/set03/03_Set03_TotalTotalUnionOfProteinsCommin
gFromLimmaAndCv.RData"
)
cb <- cb[, TotalTotalUnionCB]

###voy a coger la design matrix para cb y quitarme las muestras y
proteinas que no me interesan
designCB <-
  read.delim(
```

```
"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_CB.txt
",
    header = T,
    row.names = 1,
    sep = " "
  )
designCB <- designCB[-which(rownames(designCB) == "CYPER_M_I6_CB"), ]
rownames(designCB) <- gsub("_CB", "", rownames(designCB))
designCB <- designCB[-(which(rownames(designCB) == noestan)), ]

# Seleccion del numero de componentes para el PLS --------------------
-----
myresult <-
  pls(cb,
      clinical,
      ncomp = 25,
      mode = "regression",
      scale = FALSE)
png(
  filename = "~/Dropbox/denamic/plots/Set03/PLS/PLS_Set03_MICE_NO_CYP_
ChoosingNofComponents_.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)
par(mfrow = c(1, 2))
barplot(myresult$explained_variance$X,
        las = 2,
        main = "X")
barplot(myresult$explained_variance$Y,
        las = 2,
        main = "Y")
dev.off()

#VAMOS A UTILIZAR 2 COMPONENTES

# # Aplicacion PLS  ----------------------------------------------------
-----
# eliminando muestra unica, CYP ----------------- --------
PLS_result <-
  pls(
    X = cb,
    Y = clinical,
    scale = FALSE,
    mode = "regression",
    ncomp = 2,
    logratio = "none",
    near.zero.var = F
  )

#arreglado
par(mar = c(5, 4, 2, 2))
png(
  filename = "~/Dropbox/denamic/plots/Set03/PLS/PLS_TFG.png",
  width = 1000,
  height = 500 ,
  units = "px",
  type = "cairo",
  pointsize = 22
```

```
)

#LIMITES EJES
minimoEjesX <-
  min(c(min(PLS_result$variates[[1]]), min(PLS_result$variates[[1]])))
maximoEjesX <-
  max(c(max(PLS_result$variates[[1]]), max(PLS_result$variates[[1]])))
limitesX <- c(minimoEjesX, maximoEjesX)

minimoEjesY <-
  min(c(min(PLS_result$variates[[2]]), min(PLS_result$variates[[2]])))
maximoEjesY <-
  max(c(max(PLS_result$variates[[2]]), max(PLS_result$variates[[2]])))
limitesY <- c(minimoEjesY, maximoEjesY)

plotIndiv(
  PLS_result,
  group = factor(designCB[, 1], levels = c(
    "CAR", "CHLOR01", "CHLOR03", "CHLOR1", "VH"
  )),
  # Levels: CAR CHLOR01 CHLOR03 CHLOR1 VH
  ind.names = FALSE,
  title = "PLS Set03 ",
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designCB[, 2]],
  pch.levels = designCB[, 2],
  col.per.group = MYcolors[c("CAR", "CHLOR01", "CHLOR03", "CHLOR1",
"VH")],
  #as.character(designCB[,1])],
  size.title = rel(1.4),
  comp = c(1, 2),
  xlim = list(limitesX, limitesY),
  ylim = list(limitesX, limitesY),
  size.xlabel = rel(1.2),
  size.ylabel = rel(1.2)
)

dev.off()

# CORRELATION CIRCLE PLOT, feature selection is needed
png(
  filename =
"~/Dropbox/denamic/plots/Set03/PLS/Set03_Pls_CorrelationCirclePlot.png
",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 25
)
plotVar(object = PLS_result,
        comp = 1:2,
        cex = c(4, 10))
dev.off()

#LOADING without clinical Radial Maze
png(filename =
"~/Dropbox/denamic/plots/Set03/PLS/PLS_LoadingClinicalWithoutRadialMaz
e_Y_IMPUTED_WITH_MICE_AND_NO_CYP.png")
```

```
par(mfrow = c(1, 1))
plot(
  PLS_result$loadings$Y,
  main =
"PLSclinical_WithoutRadialMaze_Y_imputed_with_MICE_AND_NO_CYP",
  col = "red",
  xlim = c(-0.5, 1.3)
)
text(
  x = PLS_result$loadings$Y[, 1],
  y = PLS_result$loadings$Y[, 2] ,
  labels = rownames(PLS_result$loadings$Y),
  pos = 4
)
dev.off()

# CROSS-Validation, as we have a small amount of observations
#POR LOOCV
myperfLoo = perf(
  PLS_result,
  validation = "loo",
  progressBar = TRUE,
  nrepeat = 100
)

#guardado de los objetos generados y utilizados en el pls que van a
ser utilizados posteriormente en el spls...
save(cb, clinical, designCB, myperfLoo, PLS_result, file =
"~/Dropbox/denamic/data/set03/PLS_objects_for_cb_sPLS.RData")

# HP ----------------------------------------------------------------
-----
#lectura matriz X=proteomics
hp <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_HP_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
rownames(hp) <- gsub("_HP", "", rownames(hp))

cborig <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_CB_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
rownames(cborig) <- gsub("_CB", "", rownames(cborig))

#voy a ver si son exactamante las mismas ratas, las de cb que las de
hp, puesto que de esta forma
#puedo mantenter el uso de la matriz de clinical imputada y eliminar
directamente las que no se
#necesitaban en cb
sum(sort(rownames(hp)) == sort(rownames(cborig)))#27, efectivamente
por lo que puedo aplicar lo expuesto

hp <- hp[-which(rownames(hp) == "CYPER_M_I6"), ]
```

```
#puesto que hay las mismas muestras para HP que para CB
#voy a eliminar la que estaba en cb, y por ello en hp, y no en
clinical que es "CHLOR01_M_I1"
hp <- hp[-which(rownames(hp) == "CHLOR01_M_I1"), ]


#voy a ordenar la matriz hp segun el orden de cb de esta forma la
tendremos ordenada tambien igual
#que clinical
# utilizaremos la misma matriz Y imputada
# y la misma matriz de disenyo
# clinical<-clinical[rownames(cb),,drop=FALSE]
hp <- hp[rownames(cb), , drop = FALSE]
designHP <- designCB

# centrado y escalado matrices ----------------------------------------
-----
#PROTEOM
hp <- scale(hp, center = TRUE, scale = FALSE)

####CB QUEDARME SOLO CON LOS GENES CON VARIABILIDAD ELEVADA:TOTAL
TOTAL
load(

"~/Dropbox/denamic/data/set03/03_Set03_TotalTotalUnionOfProteinsCommin
gFromLimmaAndCv.RData"
)
hp <- hp[, TotalTotalUnionHP]

# Seleccion del numero de componentes para el PLS --------------------
-----
myresult <-
  pls(hp,
      clinical,
      ncomp = 24,
      mode = "regression",
      scale = FALSE)
png(
  filename = "~/Dropbox/denamic/plots/Set03/PLS/PLS_Set03_MICE_NO_CYP_
ChoosingNofComponents_HP_.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)
par(mfrow = c(1, 2))
barplot(myresult$explained_variance$X,
        las = 2,
        main = "X")
barplot(myresult$explained_variance$Y,
        las = 2,
        main = "Y")
dev.off()

#VAMOS A UTILIZAR 2 COMPONENTES

# # Aplicacion PLS  ----------------------------------------------------
-----

PLS_result <-
  pls(
```

```
    X = hp,
    Y = clinical,
    scale = FALSE,
    mode = "regression",
    ncomp = 2,
    logratio = "none",
    near.zero.var = F
  )
#arreglado
par(mar = c(5, 4, 2, 2))
png(
  filename = "~/Dropbox/denamic/plots/Set03/PLS/PLS_HP_TFG.png",
  width = 1000,
  height = 500 ,
  units = "px",
  type = "cairo",
  pointsize = 22
)

#LIMITES EJES
minimoEjesX <-
  min(c(min(PLS_result$variates[[1]]), min(PLS_result$variates[[1]])))
maximoEjesX <-
  max(c(max(PLS_result$variates[[1]]), max(PLS_result$variates[[1]])))
limitesX <- c(minimoEjesX, maximoEjesX)

minimoEjesY <-
  min(c(min(PLS_result$variates[[2]]), min(PLS_result$variates[[2]])))
maximoEjesY <-
  max(c(max(PLS_result$variates[[2]]), max(PLS_result$variates[[2]])))
limitesY <- c(minimoEjesY, maximoEjesY)


plotIndiv(
  PLS_result,
  group = factor(designHP[, 1], levels = c(
    "CAR", "CHLOR01", "CHLOR03", "CHLOR1", "VH"
  )),
  # Levels: CAR CHLOR01 CHLOR03 CHLOR1 VH
  ind.names = FALSE,
  title = "PLS Set03",
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designHP[, 2]],
  pch.levels = designHP[, 2],
  col.per.group = MYcolors[c("CAR", "CHLOR01", "CHLOR03", "CHLOR1",
"VH")],
  #as.character(designCB[,1])],
  size.title = rel(1.4),
  comp = c(1, 2),
  xlim = list(limitesX, limitesY),
  ylim = list(limitesX, limitesY),
  size.xlabel = rel(1.2),
  size.ylabel = rel(1.2)
)

dev.off()

# CORRELATION CIRCLE PLOT, feature selection is needed
png(
```

```
  filename =
"~/Dropbox/denamic/plots/Set03/PLS/Set03_Pls_HP_CorrelationCirclePlot.
png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 25
)
plotVar(object = PLS_result,
        comp = 1:2,
        cex = c(4, 10))
dev.off()

#LOADING without clinical Radial Maze
png(filename =
"~/Dropbox/denamic/plots/Set03/PLS/PLS_HP_LoadingClinicalWithoutRadial
Maze_Y_IMPUTED_WITH_MICE_AND_NO_CYP.png")
par(mfrow = c(1, 1))
plot(
  PLS_result$loadings$Y,
  main =
"PLS_HP_clinical_WithoutRadialMaze_Y_imputed_with_MICE_AND_NO_CYP",
  col = "red",
  xlim = c(-0.5, 1.3)
)
text(
  x = PLS_result$loadings$Y[, 1],
  y = PLS_result$loadings$Y[, 2] ,
  labels = rownames(PLS_result$loadings$Y),
  pos = 4
)
dev.off()

# CROSS-Validation, as we have a small amount of observations
#POR LOOCV
myperfLooHP = perf(
  PLS_result,
  validation = "loo",
  progressBar = TRUE,
  nrepeat = 100
)

PLS_result_HP <- PLS_result
clinicalHP <- clinical

#guardado de los objetos generados y utilizados en el pls que van a
ser utilizados posteriormente en el spls...
save(hp, clinicalHP, designHP, myperfLooHP, PLS_result_HP, file =
"~/Dropbox/denamic/data/set03/PLS_objects_for_HP_sPLS.RData")
```

8.5 Estudio MSEP + sPLS

```
#####################################################################
#####Estrategia 4.1 minimizacion num variables por componente####
#####################################################################
library(mixOmics)

source("~/Dropbox/denamic/scripts/Pesticides_Colors_and_GenderShapes.R
")

load("~/Dropbox/denamic/data/set03/PLS_objects_for_cb_sPLS.RData")

#############################################
####  VARIABLE SELECTION####
#############################################
Proteins_Number_1 <-
  c(
    seq(from = 5, to = 100, by = 5),
    seq(from = 100, to = 150, by = 10),
    200,
    seq(from = 250, to = 650, by = 100),
    658
  )
Proteins_Number_2 <- c(seq(from = 200, to = 400, by = 20), 658)
#X=cb, Y=clinical

matrizParaMSEP_4 <-
  matrix(ncol = length(Proteins_Number_2),
         nrow = length(Proteins_Number_1))
colnames(matrizParaMSEP_4) <- as.character(Proteins_Number_2)
rownames(matrizParaMSEP_4) <- as.character(Proteins_Number_1)

MSEP_perVariable_perProteinNumber_Matrixes_4 <-
  list(
    "MWM...Escape.Latency.Day.3" = matrizParaMSEP_4,
    "Rotarod...Time" = matrizParaMSEP_4,
    "Beam.Walking...Faults" = matrizParaMSEP_4
  )

#rows make reference to the number of variables for the C1 and columns
to the C2
for (nombre in names(MSEP_perVariable_perProteinNumber_Matrixes_4)) {
  for (numberfila in Proteins_Number_1) {
    for (numbercolumna in Proteins_Number_2) {
      SPLSresult <-
        mixOmics::spls(
          X = cb,
          Y = clinical,
          ncomp = 2,
          mode = 'regression',
          scale = FALSE,
          keepX = c(numberfila, numbercolumna),
          keepY = c(3, 3)
        )

      # LOO CV
      spls.loo <-
        perf(
          SPLSresult,
          ncomp = 2,
          mode = 'regression',
```

```
          keepX = c(numberfila, numbercolumna),
          validation = 'loo'
        )
      valorMSEP <-
        spls.loo$MSEP[which(rownames(spls.loo$MSEP) == nombre), 2]


MSEP_perVariable_perProteinNumber_Matrixes_4[[which(names(MSEP_perVari
able_perProteinNumber_Matrixes_4) ==

nombre)]][which(rownames(matrizParaMSEP_4) == numberfila),
which(colnames(matrizParaMSEP_4) ==

numbercolumna)] <- valorMSEP
    }

  }

}
MSEP_CB_perVariable_perProteinNumber_Matrixes_4 <-
  MSEP_perVariable_perProteinNumber_Matrixes_4
MSEP_CB_perVariable_perProteinNumber_Matrixes_4


#############
###PLOT 4 MWM
#############
load(

"~/Dropbox/denamic/data/set03/33MSEP_CB_perVariable_perProteinNumber_M
atrixes_4_1.RData"
)

MwmMSEP <- MSEP_CB_perVariable_perProteinNumber_Matrixes_4[[1]]
colores <-
  c(
    "orange",
    "deepskyblue1",
    "blue",
    "blue",
    "darkgreen",
    "green",
    "yellow",
    "yellow",
    "blueviolet",
    "blueviolet",
    "black",
    "red"
  )
png(
  filename =
"~/Dropbox/denamic/plots/Set03/EstrategiaSeleccionVariablesPLS/2aEstra
tegia/4_1_Intento/4_1_MSEP_CB_MWM_DependingOnTheNumberOfVariables.png"
,
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

par(mar = c(5.1, 5.1, 4.1, 2.1))
```

```
X <- as.integer(rownames(MwmMSEP))
minY <- min(MwmMSEP)
maxY <- max(MwmMSEP)
count <- 0
legend_content <- NULL

for (element in colnames(MwmMSEP)) {
  count <- count + 1
  if (count == 1) {
    plot(
      x = X,
      y = MwmMSEP[, count],
      xlab = "Number of Proteins in the C1",
      ylab = "MSEP",
      type = "l",
      col = colores[count],
      xlim = c(5, 658),
      ylim = c(minY, maxY),
      main = "MSEP_CB_MWM_DependingOnTheNumberOfVariables",
      cex.lab = 2,
      cex.axis = 1.5,
      cex.main = 2,
      lwd = 2
    )

    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
  }
  else{
    par(new = TRUE)
    plot(
      x = X,
      y = MwmMSEP[, count],
      axes = FALSE,
      xlab = "",
      ylab = "",
      type = "l",
      col = colores[count],
      ylim = c(minY, maxY),
      xlim = c(5, 658),
      lwd = 2
    )
    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
  }
}

# legend_content
legend(
  "topright",
  legend = legend_content,
  col = names(legend_content),
  fill = names(legend_content) ,
  ncol = 6,
  title = "Number of proteins in the Component 2",
  cex = 1.6,
  pt.cex = 1.3
)
abline(h = MwmMSEP[33, 12], col = "red", lwd = 2)
```

```
dev.off()

############################
#####PLOT BEAM WALKING#######
############################
BeamWalking <-
  MSEP_CB_perVariable_perProteinNumber_Matrixes_4[[which(
    names(MSEP_CB_perVariable_perProteinNumber_Matrixes_4) ==
"Beam.Walking...Faults"
  )]]
colores <-
  c(
    "orange",
    "deepskyblue1",
    "blue",
    "blue",
    "darkgreen",
    "green",
    "yellow",
    "yellow",
    "blueviolet",
    "blueviolet",
    "black",
    "red"
  )
png(
  filename =
"~/Dropbox/denamic/plots/Set03/EstrategiaSeleccionVariablesPLS/2aEstra
tegia/4_1_Intento/MSEP_CB_Beam_Walking_DependingOnTheNumberOfVariables
4.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

par(mar = c(5.1, 5.1, 4.1, 2.1))
X <- as.integer(rownames(BeamWalking))
minY <- min(BeamWalking)
maxY <- max(BeamWalking)
count <- 0
legend_content <- NULL

for (element in colnames(BeamWalking)) {
  count <- count + 1
  if (count == 1) {
    plot(
      x = X,
      y = BeamWalking[, count],
      xlab = "Number of Proteins in the C1",
      ylab = "MSEP",
      type = "l",
      col = colores[count],
      ylim = c(minY, maxY),
      xlim = c(5, 658),
      main = "MSEP_CB_Beam_Walking_DependingOnTheNumberOfVariables",
      cex.lab = 2,
      cex.axis = 1.5,
      cex.main = 2,
      lwd = 2
    )
```

```
    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
  }
  else{
    par(new = TRUE)
    plot(
      x = X,
      y = BeamWalking[, count],
      axes = FALSE,
      xlab = "",
      ylab = "",
      type = "l",
      col = colores[count],
      ylim = c(minY, maxY),
      xlim = c(5, 658),
      lwd = 2
    )
    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
  }
}

# legend_content
legend(
  "topright",
  legend = legend_content,
  col = names(legend_content),
  fill = names(legend_content) ,
  ncol = 6,
  title = "Number of proteins in the Component 2",
  cex = 1.6,
  pt.cex = 1.3
)

abline(h = BeamWalking[33, 12], col = "red", lwd = 2)

dev.off()

##################
###PLOT 4 ROTAROD
##################
RotarodMSEP <-

MSEP_CB_perVariable_perProteinNumber_Matrixes_4[[which(names(MSEP_CB_p
erVariable_perProteinNumber_Matrixes_4) ==

"Rotarod...Time")]]
colores <-
  c(
    "orange",
    "deepskyblue1",
    "blue",
    "blue",
    "darkgreen",
    "green",
    "yellow",
    "yellow",
    "blueviolet",
```

```
    "blueviolet",
    "black",
    "red"
  )
png(
  filename =
"~/Dropbox/denamic/plots/Set03/EstrategiaSeleccionVariablesPLS/2aEstra
tegia/4_1_Intento/MSEP_CB_Rotarod_DependingOnTheNumberOfVariables4_Ver
tical_ABLINE.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

par(mar = c(5.1, 5.1, 4.1, 2.1))
X <- as.integer(rownames(RotarodMSEP))
minY <- min(RotarodMSEP)
maxY <- max(RotarodMSEP)
count <- 0
legend_content <- NULL
for (element in colnames(RotarodMSEP)) {
  count <- count + 1
  if (count == 1) {
    plot(
      x = X,
      y = RotarodMSEP[, count],
      xlab = "Number of Proteins in the C1",
      ylab = "MSEP",
      type = "l",
      col = colores[count],
      ylim = c(minY, maxY),
      xlim = c(5, 658),
      main = "MSEP_CB_Rotarod_DependingOnTheNumberOfVariables",
      cex.lab = 2,
      cex.axis = 1.5,
      cex.main = 2,
      lwd = 2
    )

    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
  }
  else{
    par(new = TRUE)
    plot(
      x = X,
      y = RotarodMSEP[, count],
      axes = FALSE,
      xlab = "",
      ylab = "",
      type = "l",
      col = colores[count],
      ylim = c(minY, maxY),
      xlim = c(5, 658),
      lwd = 2
    )
    incorporacionLeyenda <- element
    names(incorporacionLeyenda) <- colores[count]
    legend_content <- c(legend_content, incorporacionLeyenda)
```

```
  }
}

# legend content
legend(
  "topright",
  legend = legend_content,
  col = names(legend_content),
  fill = names(legend_content) ,
  ncol = 6,
  title = "Number of proteins in the Component 2",
  cex = 1.6,
  pt.cex = 1.3
)

abline(
  h = RotarodMSEP[33, 12],
  col = c("red", "green"),
  lwd = 2,
  v = 65
)

dev.off()


#LIMITES EJES

####Bar plot to compare  MSEP of SPLS vs MSEP of PLS
mySPresult <-
  mixOmics::spls(
    X = cb,
    Y = clinical,
    ncomp = 2,
    mode = 'regression',
    scale = FALSE,
    keepX = c(65, 300),
    keepY = c(3, 3)
  )

SPLS_65_300_myperfLoo <-
  perf(
    mySPresult,
    ncomp = 2,
    mode = 'regression',
    keepX = c(65, 300),
    validation = 'loo'
  )

MSEP_PLS_perClinicalVariable <- myperfLoo$MSEP[, 2]
MSEP_SPLS_perClinicalVariable <- SPLS_65_300_myperfLoo$MSEP[, 2]

NombresFilas <- names(MSEP_CB_perVariable_perProteinNumber_Matrixes_4)

BarplotContent <- matrix(ncol = 2, nrow = 3)
rownames(BarplotContent) <- NombresFilas
BarplotContent <- as.data.frame(BarplotContent)
BarplotContent$MSEP_PLS <- MSEP_PLS_perClinicalVariable
BarplotContent$MSEP_SPLS_65_300 <- MSEP_SPLS_perClinicalVariable

BarplotContent <- BarplotContent[, -c(1, 2)]
```

```
BarplotContent <- as.data.frame(BarplotContent)
BarplotContent <- t(BarplotContent)
#Plots from the data in BarplotContent
# Grouped Bar Plot
png(
  filename =
"~/Dropbox/denamic/plots/Set03/EstrategiaSeleccionVariablesPLS/Set03_M
SEP_PLSvsSPLS_65_300AllClinicalVariables.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 20
)
par(mfrow = c(1, 1))
barplot(
  BarplotContent,
  main = "MSEP for each Test",
  xlab = "",
  col = c("red", "blue"),
  ylab = "MSEP 2C",
  legend = rownames(BarplotContent),
  beside = TRUE
)
dev.off()

########################
###spls application  C1=65, C2=300
########################
minEjesX <- min(mySPresult$variates[[1]])
maxEjesX <- max(mySPresult$variates[[1]])
limitEjesX <- c(minEjesX, maxEjesX)

minEjesY <- min(mySPresult$variates[[2]])
maxEjesY <- max(mySPresult$variates[[2]])
limitEjesY <- c(minEjesY, maxEjesY)


png(
  filename = "~/Dropbox/denamic/plots/Set03/SPLS/sPLS_TFG.png",
  width = 1000,
  height = 500 ,
  units = "px",
  type = "cairo",
  pointsize = 22
)
par(mfrow = c(1, 2))
plotIndiv(
  mySPresult,
  group = factor(designCB[, 1], levels = c(
    "CAR", "CHLOR01", "CHLOR03", "CHLOR1", "VH"
  )),
  # Levels: CAR CHLOR01 CHLOR03 CHLOR1 VH
  ind.names = F,
  title = "sPLS Set03 ",
  legend = F,
  style = "graphics",
  #"graphics"
  ellipse = F,
  pch = myPCH[designCB[, 2]],
```

```
  col.per.group = MYcolors[c("CAR", "CHLOR01", "CHLOR03", "CHLOR1",
"VH")],
  #as.character(designCB[,1])],
  size.title = rel(1.4),
  pch.levels = designCB[, 2],
  comp = c(1, 2),
  xlim = list(limitEjesX, limitEjesY),
  ylim = list(limitEjesX, limitEjesY),
  size.xlabel = rel(1.2),
  size.ylabel = rel(1.2)

)
dev.off()

# CORRELATION CIRCLE PLOT, feature selection is needed
png(
  filename =
"~/Dropbox/denamic/plots/Set03/SPLS/Set03_sPls_65_300_CorrelationCircl
ePlot.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 25
)
plotVar(object = mySPresult,
        comp = 1:2,
        cex = c(4, 10))
dev.off()


png(
  filename =
"~/Dropbox/denamic/plots/Set03/SPLS/NoNameSet03_sPls_CorrelationCircle
PlotNoNames.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 25
)
plotVar(
  object = mySPresult,
  comp = 1:2,
  cex = c(4, 10),
  var.names = c(TRUE, FALSE)
)
dev.off()

#correlation0.5
png(
  filename =
"~/Dropbox/denamic/plots/Set03/SPLS/03_sPls_CB_0_3_CorrelationCirclePl
otNoNames.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo",
  pointsize = 25
)
plotVar(
```

```
  object = mySPresult,
  comp = 1:2,
  cex = c(4, 10),
  var.names = c(TRUE, F),
  cutoff = 0.3,
  rad.in = 0.4
)
dev.off()

CorrelationPlotData_0_3_CB_spls <-
  plotVar(
    object = mySPresult,
    comp = 1:2,
    cex = c(4, 10),
    var.names = c(TRUE, F),
    cutoff = 0.3,
    rad.in = 0.4
  )
#
# Guardado de objeto , que contenga las N prot seleccionadas, y  que
esten separadas tambien por componente--------


TotalSelectedSPLS <- abs(mySPresult$loadings$X) > 0
# # TotalSelectedSPLS
#
table(TotalSelectedSPLS[, 1]) #65  para el primer componente
table(TotalSelectedSPLS[, 2]) #300 para el segundo componente

Componente1_Selected <-
  rownames(TotalSelectedSPLS)[TotalSelectedSPLS[, 1]]
Componente2_Selected <-
  rownames(TotalSelectedSPLS)[TotalSelectedSPLS[, 2]]
interseccion <- character(0)
for (element in Componente1_Selected) {
  if (element %in% Componente2_Selected == TRUE) {
    interseccion <- c(element, interseccion)
  }
}

CB_prot_loadings_sign <- mySPresult$loadings

save(
  Componente1_Selected,
  Componente2_Selected,
  interseccion,
  CB_prot_loadings_sign,
  file =
"~/Dropbox/denamic/data/set03/SelectedVariablesPerComponenent_and_sign
_CB_AND_Intersection4_1.RData"
)

save(CorrelationPlotData_0_3_CB_spls, file =
"~/Dropbox/denamic/data/set03/correlationPlotData_0_3_CB_spls.RData")
```

8.6 Multi-block PLS

```
################################
## multiblock  set01 #
################################
library(mixOmics)
library(ggplot2)

source("~/Dropbox/denamic/scripts/Pesticides_Colors_and_GenderShapes.R
")

# cargado matrices para multiblock
load(file =
"~/Dropbox/denamic/data/set01/MatricesParaTestsMultiBlockPLS.RData")

data = list(met = met, prot = prot)

#1.0 relacion matrices omicas
design = matrix(
  1,
  ncol = length(data),
  nrow = length(data),
  dimnames = list(names(data), names(data))
)
diag(design) = 0

# set number of component per data set
ncomp = c(2)

TCGA.block.pls = block.pls(
  X = data,
  Y = clinic,
  design = design,
  scale = F
)

# in plotindiv we color the samples per breast subtype group but the
method is unsupervised!
# here Y is the protein data set
plotIndiv(
  TCGA.block.pls,
  group = designMatrix[, 1],
  ind.names = FALSE,
  legend = T
) #funciona

#LIMITES EJES
minimoMet <-
  min(c(
    min(TCGA.block.pls$variates[[1]]),
    min(TCGA.block.pls$variates[[1]])
  ))
maximoMet <-
  max(c(
    max(TCGA.block.pls$variates[[1]]),
    max(TCGA.block.pls$variates[[1]])
  ))
limitesMet <- c(minimoMet, maximoMet)

minimoProt <-
  min(c(
```

```
    min(TCGA.block.pls$variates[[2]]),
    min(TCGA.block.pls$variates[[2]])
  ))
maximoProt <-
  max(c(
    max(TCGA.block.pls$variates[[2]]),
    max(TCGA.block.pls$variates[[2]])
  ))
limitesProt <- c(minimoProt, maximoProt)

minimoClinic <-
  min(c(
    min(TCGA.block.pls$variates[[3]]),
    min(TCGA.block.pls$variates[[3]])
  ))
maximoClinic <-
  max(c(
    max(TCGA.block.pls$variates[[3]]),
    max(TCGA.block.pls$variates[[3]])
  ))
limitesClinic <- c(minimoClinic, maximoClinic)

png(
  filename =
"~/Dropbox/denamic/plots/Set01/MultiBlockPLS/MultiBlockTFG.png",
  width = 900,
  height = 300 ,
  units = "px",
  type = "cairo",
  pointsize = 22
)
par(mfrow = c(1, 3))

plotIndiv(
  TCGA.block.pls,
  group = factor(designMatrix[, 1], levels = c("VH", "END", "CYP")),
  # Levels: CAR CHLOR01 CHLOR03 CHLOR1 VH
  ind.names = FALSE,
  title = "Multi-block PLS ",
  legend = F,
  style = "graphics",
  ellipse = F,
  pch = myPCH[designMatrix[, 2]],
  pch.levels = designMatrix[, 2],
  col.per.group = MYcolors[c("VH", "END", "CYP")],
  #as.character(designCB[,1])],
  size.title = rel(1.5),
  comp = c(1, 2),
  xlim = list(limitesMet, limitesProt, limitesClinic),
  ylim = list(limitesMet, limitesProt, limitesClinic),
  size.xlabel = rel(1.3),
  size.ylabel = rel(1.3)
)

dev.off()

save(TCGA.block.pls, file =
"~/Dropbox/denamic/data/set01/Multiblock_Result.RData")

#####################
### Loadings analysis
```

```
####################

#metabolites data
#################
LoadingsMet <- TCGA.block.pls$loadings$met
LoadingsMet <- as.data.frame(LoadingsMet)

# COMP 1
png(
  filename =
"~/Dropbox/denamic/plots/Set01/AnalisisLoadings/MetC1.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

ggplot(data = LoadingsMet, aes(abs(LoadingsMet$`comp 1`))) +
  geom_histogram(
    breaks = seq(min(abs(
      LoadingsMet$`comp 1`
    )), max(abs(
      LoadingsMet$`comp 1`
    )), by = 0.003),
    col = "red",
    fill = "blue",
    alpha = .5
  ) +
  labs(title = "Loadings Metabolomics C1") +
  labs(x = "Loading values", y = "Count") +
  theme(
    axis.text = element_text(size = 40),
    axis.title = element_text(size = 40, face = "bold"),
    title = element_text(size = 60, face = "bold")
  )

dev.off()

#Comp2
png(
  filename =
"~/Dropbox/denamic/plots/Set01/AnalisisLoadings/MetC2.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

ggplot(data = LoadingsMet, aes(abs(LoadingsMet$`comp 2`))) +
  geom_histogram(
    breaks = seq(min(abs(
      LoadingsMet$`comp 2`
    )), max(abs(
      LoadingsMet$`comp 2`
    )), by = 0.003),
    col = "red",
    fill = "blue",
    alpha = .5
  ) +
  labs(title = "Loadings Metabolomics C2") +
  labs(x = "Loading values", y = "Count") +
```

```
  theme(
    axis.text = element_text(size = 40),
    axis.title = element_text(size = 40, face = "bold"),
    title = element_text(size = 60, face = "bold")
  )

dev.off()

###############
## PROTS
###############
LoadingsProt <- TCGA.block.pls$loadings$prot
LoadingsProt <- as.data.frame(LoadingsProt)

# COMP 1
png(
  filename =
"~/Dropbox/denamic/plots/Set01/AnalisisLoadings/ProtC1.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

ggplot(data = LoadingsProt, aes(abs(LoadingsProt$`comp 1`))) +
  geom_histogram(
    breaks = seq(min(abs(
      LoadingsProt$`comp 1`
    )), max(abs(
      LoadingsProt$`comp 1`
    )), by = 0.003),
    col = "red",
    fill = "blue",
    alpha = .5
  ) +
  labs(title = "Loadings Proteomics C1") +
  labs(x = "Loading values", y = "Count") +
  theme(
    axis.text = element_text(size = 40),
    axis.title = element_text(size = 40, face = "bold"),
    title = element_text(size = 60, face = "bold")
  )

dev.off()

#Comp2
png(
  filename =
"~/Dropbox/denamic/plots/Set01/AnalisisLoadings/ProtC2.png",
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

ggplot(data = LoadingsProt, aes(abs(LoadingsProt$`comp 2`))) +
  geom_histogram(
    breaks = seq(min(abs(
      LoadingsProt$`comp 2`
    )), max(abs(
      LoadingsProt$`comp 2`
```

```
    )), by = 0.003),
    col = "red",
    fill = "blue",
    alpha = .5
  ) +
  labs(title = "Loadings Proteomics C2") +
  labs(x = "Loading values", y = "Count") +
  theme(
    axis.text = element_text(size = 40),
    axis.title = element_text(size = 40, face = "bold"),
    title = element_text(size = 60, face = "bold")
  )

dev.off()

LoadingsClinic <- TCGA.block.pls$loadings$Y
```

8.7 Selección por VIP

```
############################
## VIP study ###############
############################

library(mixOmics)

load(file = "~/Dropbox/denamic/data/set01/Multiblock_Result.RData")


##VIP function
Victor_VIP <- function(object, omic) {
  W = object$loadings[[omic]]
  H = object$ncomp
  q = ncol(object$X$Y)
  p = ncol(object$X[[omic]])
  VIP = matrix(0, nrow = p, ncol = H)
  cor2 = cor(object$X$Y, object$variates[[omic]], use = "pairwise") ^
2
  cor2 = as.matrix(cor2, nrow = q)
  VIP[, 1] = W[, 1] ^ 2
  if (H > 1) {
    for (h in 2:H) {
      if (q == 1) {
        Rd = cor2[, 1:h]
        VIP[, h] = Rd %*% t(W[, 1:h] ^ 2) / sum(Rd)
      }
      else {
        Rd = apply(cor2[, 1:h], 2, sum)
        VIP[, h] = Rd %*% t(W[, 1:h] ^ 2) / sum(Rd)
      }
    }
  }
  VIP = sqrt(p * VIP)
  rownames(VIP) = rownames(W)
  colnames(VIP) = paste("comp", 1:H)
  return(invisible(VIP))
}
prot_vip <- Victor_VIP(object = TCGA.block.pls, omic = "prot")
met_vip <- Victor_VIP(object = TCGA.block.pls, omic = "met")

# selection of metabolites with vip>1
#per component

Met_comp1 <- names(which(met_vip[, "comp 1"] > 1))
Met_comp2 <- names(which(met_vip[, "comp 2"] > 1))

#all
Met_all <- union(Met_comp1, Met_comp2)

#selection of proteins with vip>1
#per component
Prot_comp1 <- names(which(prot_vip[, "comp 1"] > 1))
Prot_comp2 <- names(which(prot_vip[, "comp 2"] > 1))

#all
Prot_all <- union(Prot_comp1, Prot_comp2)

save(Met_comp1,
  Met comp2,
  Met_all,
```

```
  Prot_comp1,
  Prot_comp2,
  Prot_all,
  file = "~/Dropbox/denamic/data/set01/VipResultsPerOmic.RData"
)
```

8.8 Correlación variables con tests

```
#########################################################################
#####
## set01, correlation variables with motor and cognitie skills per
gender #
#########################################################################
#####
library(dplyr)

load(file = "~/Dropbox/denamic/data/set01/VipResultsPerOmic.RData")

# to get the cb and clinical matrixes already used for the pls and
spls analysis, which are centered and scaled ( just clinical)
load("~/Dropbox/denamic/data/set01/MatricesParaTestsMultiBlockPLS.RDat
a")

##Proteins with Tests
ProteinsUnionComponents <- Prot_all
FeaturedCB <- prot[, ProteinsUnionComponents]

FeaturedCBSorted <- FeaturedCB
clinicalSorted <- clinic
designCBSorted <- designMatrix

##Separating Matrixes prot and clinical per gender
FeaturedCBSorted <- as.data.frame(FeaturedCBSorted)
clinicalSorted <- as.data.frame(clinicalSorted)
designCBSorted <- as.data.frame(designCBSorted)

FeaturedCBSorted$Gender <- designCBSorted$Gender
clinicalSorted$Gender <- designCBSorted$Gender

### seleccion por sexo

##seleccion   hembras
FeaturedCBSorted$ids <- rownames(FeaturedCBSorted)
clinicalSorted$ids <- rownames(clinicalSorted)

FeaturedCBSortedFemales <- FeaturedCBSorted %>% filter(Gender == "F")
rownames(FeaturedCBSortedFemales) <- FeaturedCBSortedFemales$ids
FeaturedCBSortedFemales$ids <- NULL
FeaturedCBSortedFemales$Gender <- NULL

clinicalSortedFemales <- clinicalSorted %>% filter(Gender == "F")
rownames(clinicalSortedFemales) <- clinicalSortedFemales$ids
clinicalSortedFemales$ids <- NULL
clinicalSortedFemales$Gender <- NULL

###seleccion machos
FeaturedCBSortedMales <- FeaturedCBSorted %>% filter(Gender == "M")
rownames(FeaturedCBSortedMales) <- FeaturedCBSortedMales$ids
FeaturedCBSortedMales$ids <- NULL
FeaturedCBSortedMales$Gender <- NULL

clinicalSortedMales <- clinicalSorted %>% filter(Gender == "M")
rownames(clinicalSortedMales) <- clinicalSortedMales$ids
clinicalSortedMales$ids <- NULL
clinicalSortedMales$Gender <- NULL

##correlation study
```

```
nombrescolumnas <- c(colnames(clinic), "Most_Correlated", "Value")

CorrelationDataFrame <- matrix(data = rep(0),
                               ncol = 7,
                               nrow = 265)
colnames(CorrelationDataFrame) <- nombrescolumnas
rownames(CorrelationDataFrame) <- ProteinsUnionComponents

CorrelationDataFrame <- as.data.frame(CorrelationDataFrame)

#################
#Males
#FOR PEARSON
#################
CorrelationDataFramePearsonMales <- CorrelationDataFrame

##filling the matrix,3 first columns
for (protein in ProteinsUnionComponents) {
  for (ClinicalTest in colnames(clinicalSortedMales)) {
    Corr <-
      cor(x = FeaturedCBSortedMales[, protein],
          y = clinicalSortedMales[, ClinicalTest],
          method = "pearson")
    CorrelationDataFramePearsonMales[protein, ClinicalTest] <- Corr
  }
}

#filling the last two columns
CorrelationDataFramePearsonMales <-
  as.data.frame(CorrelationDataFramePearsonMales)

for (Line in 1:nrow(CorrelationDataFramePearsonMales)) {
  TestPos <-
    which(abs(CorrelationDataFramePearsonMales[Line, ][1:5]) ==
max(abs(CorrelationDataFramePearsonMales[Line, ][1:5])))
  Test <- colnames(CorrelationDataFramePearsonMales)[TestPos]
  Value <- CorrelationDataFramePearsonMales[Line, TestPos]
  CorrelationDataFramePearsonMales[Line, "Most_Correlated"] <- Test
  CorrelationDataFramePearsonMales[Line, "Value"] <- Value
}

# Grouping the proteins by most correlated tests
PearsonM <- CorrelationDataFramePearsonMales
PearsonM <- PearsonM[order(PearsonM$Most_Correlated), ]

##### MOTOR TESTS
#BEAM WALKING
PearsonBeamWalkingM <- PearsonM
PearsonBeamWalkingM$Prots <- rownames(PearsonM)
PearsonBeamWalkingM <-
  PearsonBeamWalkingM %>% filter(Most_Correlated ==
"Beam.Walking...Faults")
rownames(PearsonBeamWalkingM) <- PearsonBeamWalkingM$Prots
PearsonBeamWalkingM$Prots <- NULL
#ROTAROD
PearsonRotarodM <- PearsonM
PearsonRotarodM$Prots <- rownames(PearsonM)
PearsonRotarodM <-
  PearsonRotarodM %>% filter(Most_Correlated == "Rotarod...Time")
rownames(PearsonRotarodM) <- PearsonRotarodM$Prots
PearsonRotarodM$Prots <- NULL
```

```
#### COGNITIVE TESTS
#MWM
PearsonMWMM <- PearsonM
PearsonMWMM$Prots <- rownames(PearsonM)
PearsonMWMM <-
  PearsonMWMM %>% filter(Most_Correlated ==
"MWM...Escape.Latency.Day.3")
rownames(PearsonMWMM) <- PearsonMWMM$Prots
PearsonMWMM$Prots <- NULL

save(PearsonM,
     PearsonBeamWalkingM,
     PearsonRotarodM,
     PearsonMWMM,
     file =
"~/Dropbox/denamic/data/set03/06_DonePerGender_Males_CorrelatedProtein
sWithTests_ForFunctionalEnrichment.RData")

#################
#Females
#FOR PEARSON
#################
CorrelationDataFramePearsonFemales <- CorrelationDataFrame

##filling the matrix,3 first columns
for (protein in ProteinsUnionComponents) {
  for (ClinicalTest in colnames(clinicalSortedFemales)) {
    Corr <-
      cor(x = FeaturedCBSortedFemales[, protein],
          y = clinicalSortedFemales[, ClinicalTest],
          method = "pearson")
    CorrelationDataFramePearsonFemales[protein, ClinicalTest] <- Corr
  }
}

#filling the last two columns
CorrelationDataFramePearsonFemales <-
  as.data.frame(CorrelationDataFramePearsonFemales)

for (Line in 1:nrow(CorrelationDataFramePearsonFemales)) {
  TestPos <-
    which(abs(CorrelationDataFramePearsonFemales[Line, ][1:3]) ==
max(abs(CorrelationDataFramePearsonFemales[Line, ][1:3])))
  Test <- colnames(CorrelationDataFramePearsonFemales)[TestPos]
  Value <- CorrelationDataFramePearsonFemales[Line, TestPos]
  CorrelationDataFramePearsonFemales[Line, "Most_Correlated"] <- Test
  CorrelationDataFramePearsonFemales[Line, "Value"] <- Value

}

# Grouping the proteins by most correlated tests
PearsonF <- CorrelationDataFramePearsonFemales
PearsonF <- PearsonF[order(PearsonF$Most_Correlated), ]

##### MOTOR TESTS
#BEAM WALKING
PearsonBeamWalkingF <- PearsonF
PearsonBeamWalkingF$Prots <- rownames(PearsonF)
PearsonBeamWalkingF <-
```

```
  PearsonBeamWalkingF %>% filter(Most_Correlated ==
"Beam.Walking...Faults")
rownames(PearsonBeamWalkingF) <- PearsonBeamWalkingF$Prots
PearsonBeamWalkingF$Prots <- NULL

#ROTAROD
PearsonRotarodF <- PearsonF
PearsonRotarodF$Prots <- rownames(PearsonF)
PearsonRotarodF <-
  PearsonRotarodF %>% filter(Most_Correlated == "Rotarod...Time")
rownames(PearsonRotarodF) <- PearsonRotarodF$Prots
PearsonRotarodF$Prots <- NULL

#### COGNITIVE TESTS
#MWM
PearsonMWMF <- PearsonF
PearsonMWMF$Prots <- rownames(PearsonF)
PearsonMWMF <-
  PearsonMWMF %>% filter(Most_Correlated ==
"MWM...Escape.Latency.Day.3")
rownames(PearsonMWMF) <- PearsonMWMF$Prots
PearsonMWMF$Prots <- NULL

save(PearsonF,
     PearsonBeamWalkingF,
     PearsonRotarodF,
     PearsonMWMF,
     file =
"~/Dropbox/denamic/data/set03/06_DonePerGender_Females_CorrelatedProte
insWithTests_ForFunctionalEnrichment.RData")
###metab with Tests
```

8.9 Enriquecimiento funcional

```
#########################################################################
##############################################
## Per Gender Functional Enrichment Proteins correlated positvively
and negatively with Motor and Cogntivie skills #
#########################################################################
##############################################
library(biomaRt)
library(ggplot2)

#Functions For the Functional Enrichment Analysis
EnrichALLterms = function (test,
                           notTest,
                           annotation,
                           p.adjust.method = "fdr") {
  annot2test = unique(annotation[, 2])
  #IN TEST THE DIFFERENTIALLY expressed genes, in not test the rest of
genes
  resultat = t(
    sapply(
      annot2test,
      Enrich1term,
      test = test,
      notTest = notTest,
      annotation = annotation
    )
  )

  return (data.frame(
    resultat,
    "adjPval" = p.adjust(as.numeric(resultat[, "pval"]), method =
p.adjust.method),
    stringsAsFactors = F
  ))

}

Enrich1term = function (term, test, notTest, annotation) {
  annotTest = length(intersect(test, annotation[annotation[, 2] ==
term, 1]))

  if ((annotTest) > 0) {
    annotNOTtest = length(intersect(notTest, annotation[annotation[,
2] == term, 1]))
    mytest = matrix(c(
      annotTest,
      length(test) - annotTest,
      annotNOTtest,
      length(notTest) - annotNOTtest
    ),
    ncol = 2)
    resultat = c(
      term,
      annotTest,
      length(test),
      annotNOTtest,
      length(notTest),
      fisher.test(mytest, alternative = "greater")$p.value
    )
    names(resultat) = c("term",
```

```
                          "annotTest",
                          "test",
                          "annotNotTest",
                          "notTest",
                          "pval")
  } else {
    resultat = c(term, 0, 0, 0, 0, 100)
    names(resultat) = c("term",
                          "annotTest",
                          "test",
                          "annotNotTest",
                          "notTest",
                          "pval")
  }

  return(resultat)

}
##Preparacion fichero anotacion
biomartRnorvegicus = useMart(biomart = "ENSEMBL_MART_ENSEMBL",
                             dataset = "rnorvegicus_gene_ensembl",
                             host = "dec2016.archive.ensembl.org")
atributos = listAttributes(biomartRnorvegicus)

#FULL PROT = 1223 PROT DEL ANALISIS POST NA SELECCION
FULLPROTCB <-
  read.delim(
    "~/Dropbox/denamic/data/set03/01_Set03_CB_Imputada_Por_MICE.txt",
    header = TRUE,
    row.names = 1,
    sep = " "
  )

totesprotCB <- colnames(FULLPROTCB)

myannotTOTALCBbiomart = getBM(
  attributes = c("uniprot_swissprot", "uniprot_genename",
"name_1006"),
  filters = "uniprot_swissprot" ,
  values = totesprotCB,
  mart = biomartRnorvegicus
)

#####################
#MALES
#####################

load(file =
"~/Dropbox/denamic/data/set03/06_DonePerGender_Males_CorrelatedProtein
sWithTests_ForFunctionalEnrichment.RData")

###FUNCTIONAL ENRICHMENT CON LAS PROTEINAS CORRELACIONADAS
POSITIVAMENTE CON LA MEJORA MOTORA

# Las correlacionadas positivamente con Rotarod y Negativamente con
Beam Walking
MejoraMotoraProtsMales <-
  c(rownames(PearsonRotarodM)[PearsonRotarodM$Value > 0],
rownames(PearsonBeamWalkingM)[PearsonBeamWalkingM$Value < 0])

myEnrichResultsMejoraMotoraMales = EnrichALLterms(
```

```
  test = MejoraMotoraProtsMales,
  notTest = setdiff(totesprotCB, MejoraMotoraProtsMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

### CORR POSITIVA ROTAROD
CorrPosRotarodMales <-
  rownames(PearsonRotarodM)[PearsonRotarodM$Value > 0]

FECorrPosRotarodMales = EnrichALLterms(
  test = CorrPosRotarodMales,
  notTest = setdiff(totesprotCB, CorrPosRotarodMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

##CORR NEGATIVA ROTAROD
CorrNegRotarodMales <-
  rownames(PearsonRotarodM)[PearsonRotarodM$Value < 0]

FECorrNegRotarodMales = EnrichALLterms(
  test = CorrNegRotarodMales,
  notTest = setdiff(totesprotCB, CorrNegRotarodMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

##CORR POSITIVA BEAM WALKING
CorrPosBWMales <-
  rownames(PearsonBeamWalkingM)[PearsonBeamWalkingM$Value > 0]

FECorrPosBWMales = EnrichALLterms(
  test = CorrPosBWMales,
  notTest = setdiff(totesprotCB, CorrPosBWMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)
View(FECorrPosBWMales)

## CORR NEGATIVA BEAM WALKING

CorrNegBWMales <-
  rownames(PearsonBeamWalkingM)[PearsonBeamWalkingM$Value < 0]

FECorrNegBWMales = EnrichALLterms(
  test = CorrNegBWMales,
  notTest = setdiff(totesprotCB, CorrNegBWMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

#######################################
##Mejora motora corr values histogram machos
# Beam Walking
PearsonMejoraMotoraMales <- PearsonM[MejoraMotoraProtsMales, ]

png(
  filename =
"~/Dropbox/denamic/plots/Set03/AnalisisFuncionalCB/HistogramsCorrelati
on/MejoraMotoraMachos.png",
```

```
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

ggplot(data = PearsonMejoraMotoraMales,
aes(PearsonMejoraMotoraMales$Value)) +
  geom_histogram(
    breaks = seq(
      min(PearsonMejoraMotoraMales$Value),
      max(PearsonMejoraMotoraMales$Value),
      by = 0.05
    ),
    col = "red",
    fill = "blue",
    alpha = .5
  ) +
  labs(title = "Mejora Motora Machos") +
  labs(x = "Correlation Values", y = "Count") +
  theme(
    axis.text = element_text(size = 40),
    axis.title = element_text(size = 40, face = "bold"),
    title = element_text(size = 60, face = "bold")
  )

dev.off()

###FUNCTIONAL ENRICHMENT CON LAS PROTEINAS CORRELACIONADAS
NEGATIVAMENTE CON LA MEJORA MOTORA

# Las correlacionadas negativamente con Rotarod y positivamente con
Beam Walking
EmpeoranMotoraProtsMales <-
  c(rownames(PearsonRotarodM)[PearsonRotarodM$Value < 0],
rownames(PearsonBeamWalkingM)[PearsonBeamWalkingM$Value > 0])


myEnrichResultsEmpeoranMotoraMales = EnrichALLterms(
  test = EmpeoranMotoraProtsMales,
  notTest = setdiff(totesprotCB, EmpeoranMotoraProtsMales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)


# MWM, FE ENRICHMENT CON LAS CORRELACIONADAS NEGATIVAMENTE CON LA
MEJORA COGNITIVA
## COR positiva con MWM
MWMPosM <- rownames(PearsonMWMM)[PearsonMWMM$Value > 0]

myEnrichResultsMWMEmpeoranCognitivoMales = EnrichALLterms(
  test = MWMPosM,
  notTest = setdiff(totesprotCB, MWMPosM),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

##FE ENRICHMENT CON LAS CORRELACIONADAS POSITIVAMENTE CON LA MEJORA
COGNITIVA
#cor negativa con MWM
```

```
MWMNegM <- rownames(PearsonMWMM)[PearsonMWMM$Value < 0]

myEnrichResultsMWMMejoraCognitivaMales = EnrichALLterms(
  test = MWMNegM,
  notTest = setdiff(totesprotCB, MWMNegM),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

save(
  myEnrichResultsEmpeoranMotoraMales,
  myEnrichResultsMejoraMotoraMales,
  myEnrichResultsMWMEmpeoranCognitivoMales,
  myEnrichResultsMWMMejoraCognitivaMales,
  file =
"~/Dropbox/denamic/data/set03/FEnrichment_PerGender_Males_Results_MejoraYEmpeoraCognitivoYMotora.RData"
)


##########
## FEMALES
##########

load(file =
"~/Dropbox/denamic/data/set03/06_DonePerGender_Females_CorrelatedProteinsWithTests_ForFunctionalEnrichment.RData")

### CORR POSITIVA ROTAROD

CorrPosRotarodFemales <-
  rownames(PearsonRotarodF)[PearsonRotarodF$Value > 0]

FECorrPosRotarodFemales = EnrichALLterms(
  test = CorrPosRotarodFemales,
  notTest = setdiff(totesprotCB, CorrPosRotarodFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

##CORR NEGATIVA ROTAROD
CorrNegRotarodFemales <-
  rownames(PearsonRotarodF)[PearsonRotarodF$Value < 0]

FECorrNegRotarodFemales = EnrichALLterms(
  test = CorrNegRotarodFemales,
  notTest = setdiff(totesprotCB, CorrNegRotarodFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)


##CORR POSITIVA BEAM WALKING
CorrPosBWFemales <-
  rownames(PearsonBeamWalkingF)[PearsonBeamWalkingF$Value > 0]

FECorrPosBWFemales = EnrichALLterms(
  test = CorrPosBWFemales,
  notTest = setdiff(totesprotCB, CorrPosBWFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
```

```
)

## CORR NEGATIVA BEAM WALKING

CorrNegBWFemales <-
  rownames(PearsonBeamWalkingF)[PearsonBeamWalkingF$Value < 0]

FECorrNegBWFemales = EnrichALLterms(
  test = CorrNegBWFemales,
  notTest = setdiff(totesprotCB, CorrNegBWFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

###FUNCTIONAL ENRICHMENT CON LAS PROTEINAS CORRELACIONADAS
POSITIVAMENTE CON LA MEJORA MOTORA

# Las correlacionadas positivamente con Rotarod y Negativamente con
Beam Walking

MejoraMotoraProtsFemales <-
  c(rownames(PearsonRotarodF)[PearsonRotarodF$Value > 0],
rownames(PearsonBeamWalkingF)[PearsonBeamWalkingF$Value <

0])

myEnrichResultsMejoraMotoraFemales = EnrichALLterms(
  test = MejoraMotoraProtsFemales,
  notTest = setdiff(totesprotCB, MejoraMotoraProtsFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

###FUNCTIONAL ENRICHMENT CON LAS PROTEINAS CORRELACIONADAS
NEGATIVAMENTE CON LA MEJORA MOTORA

# Las correlacionadas negativamente con Rotarod y positivamente con
Beam Walking

EmpeoranMotoraProtsFemales <-
  c(rownames(PearsonRotarodF)[PearsonRotarodF$Value < 0],
rownames(PearsonBeamWalkingF)[PearsonBeamWalkingF$Value >

0])

myEnrichResultsEmpeoranMotoraFemales = EnrichALLterms(
  test = EmpeoranMotoraProtsFemales,
  notTest = setdiff(totesprotCB, EmpeoranMotoraProtsFemales),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)


# MWM, FE ENRICHMENT CON LAS CORRELACIONADAS NEGATIVAMENTE CON LA
MEJORA COGNITIVA
## COR positiva con MWM

MWMPosF <- rownames(PearsonMWMF)[PearsonMWMF$Value > 0]

myEnrichResultsMWMEmpeoranCognitivoFemales = EnrichALLterms(
  test = MWMPosF,
```

```
  notTest = setdiff(totesprotCB, MWMPosF),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

##FE ENRICHMENT CON LAS CORRELACIONADAS POSITIVAMENTE CON LA MEJORA
COGNITIVA
#cor negativa con MWM
MWMNegF <- rownames(PearsonMWMF)[PearsonMWMF$Value < 0]

myEnrichResultsMWMMejoraCognitivaFemales = EnrichALLterms(
  test = MWMNegF,
  notTest = setdiff(totesprotCB, MWMNegF),
  annotation = myannotTOTALCBbiomart[, c(1, 3)],
  p.adjust.method = "fdr"
)

save(
  myEnrichResultsEmpeoranMotoraFemales,
  myEnrichResultsMejoraMotoraFemales,
  myEnrichResultsMWMEmpeoranCognitivoFemales,
  myEnrichResultsMWMMejoraCognitivaFemales,
  file =
"~/Dropbox/denamic/data/set03/FEnrichment_PerGender_Females_Results_Me
joraYEmpeoraCognitivoYMotora.RData"
)
```

8.10 Estudio del comportamiento, promediado, de las variables

```
#####################################
## Set01 Function Protein Behaviour #
#####################################
library(dplyr)

##seguir del load matriu promediada
load(file =
"~/Dropbox/denamic/data/set01/Set01_ProtPromediada_sinCentradoNiEscala
do.RData")

clinicalTodasImputada <-
  read.delim(file =
"~/Dropbox/denamic/data/set01/00_Set01_prot_CB.txt",
             header = T,
             row.names = 1)

clinicalTodasImputada <- t(clinicalTodasImputada)
rownames(clinicalTodasImputada) <-
  gsub(
    pattern = ".",
    replacement = "_",
    rownames(clinicalTodasImputada),
    fixed = T
  )
rownames(clinicalTodasImputada) <-
  gsub(
    pattern = "_CB",
    replacement = "",
    rownames(clinicalTodasImputada),
    fixed = T
  )
clinicalTodasImputada <- as.data.frame(clinicalTodasImputada)

clinicalPromediada <- ProtPromediada
rownames(clinicalPromediada) <- clinicalPromediada$names
clinicalPromediada$names <- NULL
clinicalPromediada <- t(clinicalPromediada)
clinicalPromediada <- as.data.frame(clinicalPromediada)

TestBehaviour <- function(protein) {
  MatrizPromediadaCB <- clinicalPromediada
  CB <- clinicalTodasImputada
  cbALL <- CB

  protinteresPROMEDIADA <- MatrizPromediadaCB[protein, , drop = F]

  Xvalues <- 1:(length(colnames(protinteresPROMEDIADA)) / 2)

  protintFemales <-
    protinteresPROMEDIADA[, c("VH_F", "CYP_F", "END_F"), drop = F]
  protintMales <-
    protinteresPROMEDIADA[, c("VH_M", "CYP_M", "END_M"), drop = F]

  ####To plot SE, groups
```

```
  ##afegir el Standard Error per punt
  CB <- CB[sort(rownames(CB)), protein, drop = F]

  designCB <- do.call("rbind", strsplit(row.names(CB), "_", fixed =
T))
  designCB <- designCB[, -c(3, 4)]
  designCB <- as.data.frame(designCB)
  colnames(designCB) <- c("Treatment", "Gender")

  CB$Gender <- designCB$Gender
  CB$Treatment <- designCB$Treatment

  ###TO PLOT ALL LINE
  cbALL <- cbALL[sort(rownames(cbALL)), protein, drop = F]

  designCBALL <-
    do.call("rbind", strsplit(row.names(cbALL), "_", fixed = T))
  designCBALL <- designCBALL[, 1]
  designCBALL <- as.data.frame(designCBALL)
  colnames(designCBALL) <- "Treatment"

  cbALL$Treatment <- designCBALL$Treatment

  ##Selection by gender and Treatment, groups

  #For genders line
  CB$names <- rownames(CB)

  #Males VH
  MalesVH <- CB %>% filter(Gender == "M", Treatment == "VH")
  #Females vH
  FemalesVH <- CB %>% filter(Gender == "F", Treatment == "VH")

  #Males CAR
  MalesEnd <- CB %>% filter(Gender == "M", Treatment == "END")

  #Females CAR
  FemalesEnd <- CB %>% filter(Gender == "F", Treatment == "END")

  #Males CHLOR01
  MalesCyp <- CB %>% filter(Gender == "M", Treatment == "CYP")

  #Females CHLORO1
  FemalesCyp <- CB %>% filter(Gender == "F", Treatment == "CYP")

  PestsGenderF <-
    list(
      "FemalesVH" = FemalesVH,
      "FemalesCyp" = FemalesCyp,
      "FemalesEnd" = FemalesEnd
    )

  PestsGenderM <-
    list("MalesVH" = MalesVH,
         "MalesCyp" = MalesCyp,
         "MalesEnd" = MalesEnd)

  ###For ALL line
  cbALL$names <- rownames(cbALL)

  # VH
```

```
VH <- cbALL %>% filter(Treatment == "VH")

# CAR
End <- cbALL %>% filter(Treatment == "END")

# CHLOR01
Cyp <- cbALL %>% filter(Treatment == "CYP")

Pests <- list("VH" = VH,
              "Cyp" = Cyp,
              "End" = End)

###PLOT PARA LOS PROMEDIOS
minY <- min(CB[, protein])
maxY <- max(CB[, protein])

route <-
  paste("~/Dropbox/denamic/plots/Set01/AnalisisFuncionalCB/",
        protein,
        ".png",
        sep = "")
png(
  filename = route,
  width = 1280,
  height = 720 ,
  units = "px",
  type = "cairo"
)

par(mar = c(5.1, 5.1, 4.1, 2.1))

# for All line, Y values
YvaluesAll <- NULL

for (group in Pests) {
  YvaluesAll <- c(YvaluesAll, (mean(group[, protein])))
}

#Females
plot(
  x = Xvalues,
  y = protintFemales[1, ],
  type = "l",
  col = "black",
  main = protein,
  cex.lab = 2,
  cex.axis = 1.2,
  cex.main = 2,
  lwd = 2,
  xlim = c(1, 3),
  ylim = c(minY, maxY),
  xaxt = "n",
  ylab = "Averaged Expression",
  xlab = "Condition"
)
axis(
  1,
  at = 1:3,
  labels = c("VH", "CYP", "END"),
  cex.axis = 1.5
)
```

```r
#Standard Errors
pos <- 0
for (Treatment in PestsGenderF) {
  pos <- pos + 1
  Barra <- Treatment[, protein]
  MeanBarra <- mean(Barra)
  SEBarra <- sd(Barra) / length(Barra)
  minBarra <- MeanBarra - SEBarra
  maxBarra <- MeanBarra + SEBarra
  par(new = TRUE)
  plot(
    x = c(pos, pos),
    y = c(minBarra, maxBarra),
    axes = FALSE,
    xlab = "",
    ylab = "",
    type = "l",
    col = "black",
    ylim = c(minY, maxY),
    xlim = c(1, 3),
    lwd = 1
  )
}

#Males
par(new = TRUE)
plot(
  x = Xvalues,
  y = protintMales[1, ],
  axes = FALSE,
  xlab = "",
  ylab = "",
  type = "l",
  col = "blue",
  ylim = c(minY, maxY),
  xlim = c(1, 3),
  lwd = 2
)

pos <- 0
for (Treatment in PestsGenderM) {
  pos <- pos + 1
  #SEdata<-
  Barra <- Treatment[, protein]
  MeanBarra <- mean(Barra)
  SEBarra <- sd(Barra) / length(Barra)
  minBarra <- MeanBarra - SEBarra
  maxBarra <- MeanBarra + SEBarra
  par(new = TRUE)
  plot(
    x = c(pos, pos),
    y = c(minBarra, maxBarra),
    axes = FALSE,
    xlab = "",
    ylab = "",
    type = "l",
    col = "blue",
    ylim = c(minY, maxY),
    xlim = c(1, 3),
    lwd = 1
```

```
    )
  }

  #ALL line
  par(new = TRUE)

  plot(
    x = Xvalues,
    y = YvaluesAll,
    axes = FALSE,
    lty = 2,
    xlab = "",
    ylab = "",
    type = "l",
    col = "red",
    xlim = c(1, 3),
    ylim = c(minY, maxY),
    lwd = 2
  )

  #ALL Standard Error
  pos <- 0
  for (group in Pests) {
    pos <- pos + 1
    Barra <- group[, protein]
    MeanBarra <- mean(Barra)
    SEBarra <- sd(Barra) / length(Barra)
    minBarra <- MeanBarra - SEBarra
    maxBarra <- MeanBarra + SEBarra
    par(new = TRUE)
    plot(
      x = c(pos, pos),
      y = c(minBarra, maxBarra),
      axes = FALSE,
      xlab = "",
      ylab = "",
      type = "l",
      col = "red",
      ylim = c(minY, maxY),
      xlim = c(1, 3),
      lwd = 1
    )
  }

  legend(
    "topright",
    legend = c("Females", "Males", "All"),
    col = c("black", "blue", "red"),
    fill = c("black", "blue", "red"),
    ncol = 1,
    title = "Gender",
    cex = 1.6
  )

  dev.off()
}

# Example with P21707
TestBehaviour("P21707")
which(rownames(clinicalPromediada) == "P21707")
```

8.11 Filtrado por *limma y CV*

```
########################################
#### SET03: filtering low variability ####
########################################
library(dplyr)
library(limma)

# CB ------------------------------------------------------------------
-----

# #Preparacion de matriz con una columna de nombres que incluya la
combinación del  pesticida y el sexo--------

CB <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_CB_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminacion rata tratada con CYP ya que solo hay una
CB <- CB[-which(rownames(CB) == "CYPER_M_I6_CB"), ]
CB2 <- CB

designCB <- do.call("rbind", strsplit(row.names(CB), "_", fixed = T))
designCB <- designCB[, -c(3, 4)]
data <- designCB
data <- as.data.frame(data)
data$Combination <-
  apply(data[, c(1, 2)] , 1 , paste , collapse = "_")

CB2$names <- data$Combination

new_CB <- CB2 %>% select(names, everything())

# Obtencion de la matriz que tenga promediada la expresion proteica
por pesticida y sexo --------

matriz_promediada_CB <- aggregate(. ~ names, FUN = mean, data =
new_CB)

#comprobacion de que esta bien promediado
#para CAR_F la tabla me dice que la media es de 0.52877666 ( para la
prot A1L1J8)
#manualmente: para CAR_F me sale (0.113760115 + 0.943793213)/2=
0.5287767 , perfecto pues.

# Guardado de matriz promediada ---------------------------------------
-----

matriz_promediada_CB <- t(matriz_promediada_CB)
colnames(matriz_promediada_CB) <- matriz_promediada_CB[1, ]
matriz_promediada_CB <- matriz_promediada_CB[-1, ]

write.table(
  matriz_promediada_CB,
  file =
"~/Dropbox/denamic/data/set03/02_Set03_CB_Matriz_Promediada_Por_Pestic
ida_Y_Sexo.txt",
  sep = "\t",
```

```
  quote = F,
  col.names = T,
  row.names = T
)

# HP ----------------------------------------------------------------
-----
# #Preparacion de matriz con una columna de nombres que incluya la
combinación del  pesticida y el sexo--------
HP <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_HP_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminacion rata tratada con CYP ya que solo hay una
HP <- HP[-which(rownames(HP) == "CYPER_M_I6_HP"), ]
HP2 <- HP

designHP <- do.call("rbind", strsplit(row.names(HP), "_", fixed = T))
designHP <- designHP[, -c(3, 4)]
data <- designHP
data <- as.data.frame(data)
data$Combination <-
  apply(data[, c(1, 2)] , 1 , paste , collapse = "_")

HP2$names <- data$Combination

new_HP <- HP2 %>% select(names, everything())

# Obtencion de la matriz que tenga promediada la expresion proteica
por pesticida y sexo --------

matriz_promediada_HP <- aggregate(. ~ names, FUN = mean, data =
new_HP)

#comprobacion de que esta bien promediado
#para CAR_F la tabla me dice que la media es de 0.52877666 ( para la
prot A1L1J8)
#manualmente: para CAR_F me sale (0.113760115 + 0.943793213)/2=
0.5287767 , perfecto pues.

# Guardado de matriz promediada -------------------------------------
-----

matriz_promediada_HP <- t(matriz_promediada_HP)
colnames(matriz_promediada_HP) <- matriz_promediada_HP[1, ]
matriz_promediada_HP <- matriz_promediada_HP[-1, ]

write.table(
  matriz_promediada_HP,
  file =
"~/Dropbox/denamic/data/set03/02_Set03_HP_Matriz_Promediada_Por_Pestic
ida_Y_Sexo.txt",
  sep = "\t",
  quote = F,
  col.names = T,
  row.names = T
)
```

```r
# Deteccion de proteinas con baja variabilidad ----------------------
-----
matriz_promediada_HP <-
  read.delim(

"~/Dropbox/denamic/data/set03/02_Set03_HP_Matriz_Promediada_Por_Pestic
ida_Y_Sexo.txt",
    header = T,
    sep = "\t",
    row.names = 1
  )
matriz_promediada_CB <-
  read.delim(

"~/Dropbox/denamic/data/set03/02_Set03_CB_Matriz_Promediada_Por_Pestic
ida_Y_Sexo.txt",
    header = T,
    sep = "\t",
    row.names = 1
  )

# CB ----------------------------------------------------------------
-----

sdCB <- apply(matriz_promediada_CB, 1, sd)
# boxplot(sdCB)

meanCB = rowMeans(matriz_promediada_CB)

cvCB = abs(sdCB / meanCB)

cvCB_3 <- names(which(cvCB > 3))#587

# HP ----------------------------------------------------------------
-----

sdHP <- apply(matriz_promediada_HP, 1, sd)

meanHP = rowMeans(matriz_promediada_HP)

cvHP = abs(sdHP / meanHP)

sum(cvHP > 3)#477 prot

cvHP_3 <- names(which(cvHP > 3))


# Limma analysis ----------------------------------------------------
-----

designHP <-
  read.delim(

"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_HP.txt
",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminamos CYP
designHP <- designHP[-which(rownames(designHP) == "CYPER_M_I6_HP"), ]
```

```
designCB <-
  read.delim(

"~/Dropbox/denamic/data/set03/DesignMatrixes/designProt03_ARSYN_CB.txt
",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminamos CYP
designCB <- designCB[-which(rownames(designCB) == "CYPER_M_I6_CB"), ]


# CB ----------------------------------------------------------------
-----

#Expresion matrixes
protCB <-
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_CB_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminacion CYP
protCB <- protCB[-which(rownames(protCB) == "CYPER_M_I6_CB"), ]

# #voom must be used if we need to log transf the data

sex_cb <- factor(designCB[rownames(protCB), "Gender"])

pest_cb = factor(designCB[rownames(protCB), "Pesticide"], levels =
c("VH", "CHLOR01", "CHLOR03", "CHLOR1", "CAR"))

prot_matrix_CB = model.matrix( ~ sex_cb + pest_cb + sex_cb * pest_cb)

fit_CB = lmFit(t(protCB), prot_matrix_CB)

fit_p <- eBayes(fit_CB)

fit_p$p.value

# TopRankedCB -------------------------------------------------------
-----

betas <-
  colnames(fit_p$p.value)[-c(1, 2)] #quitamos intercept y  la que solo
estudia el efecto del sexo no nos interesan
TopRankedCBLists <- list()

for (element in betas) {
  TopRankedCBLists[[element]] <-
    topTable(fit_p, coef = element, number = 1223)
}


pvaluesNumber_CB_Signif <- list()
padjNumber_CB_Signif <- list()

pvaluesNames_CB_Signif <- list()
```

```
padjNames_CB_Signif <- list()

for (position in 1:length(TopRankedCBLists)) {
  pvaluesNames_CB_Signif[[names(TopRankedCBLists[position])]] <-
    character()

  padjNames_CB_Signif[[names(TopRankedCBLists[position])]] <-
    character()

  for (row in 1:nrow(TopRankedCBLists[[position]])) {
    if (TopRankedCBLists[[position]][row, 4] < 0.05) {
      pvaluesNames_CB_Signif[[names(TopRankedCBLists[position])]] <-
        c(pvaluesNames_CB_Signif[[names(TopRankedCBLists[position])]],
rownames(TopRankedCBLists[[position]][row, ]))
    }
  }

  pvaluesNumber_CB_Signif[[names(TopRankedCBLists[position])]] <-

length(pvaluesNames_CB_Signif[[names(TopRankedCBLists[position])]])

  for (row in 1:nrow(TopRankedCBLists[[position]])) {
    if (TopRankedCBLists[[position]][row, 5] < 0.05) {
      padjNames_CB_Signif[[names(TopRankedCBLists[position])]] <-
        c(padjNames_CB_Signif[[names(TopRankedCBLists[position])]],
rownames(TopRankedCBLists[[position]][row, ]))
    }
  }

  padjNumber_CB_Signif[[names(TopRankedCBLists[position])]] <-
    length(padjNames_CB_Signif[[names(TopRankedCBLists[position])]])
}
pvaluesNumber_CB_Signif
padjNumber_CB_Signif
# Unions-Unique analysis ----------------------------------------------
------------

PCBlist <-
  list("pvaluesNames_CB_Signif" = pvaluesNames_CB_Signif,
       "padjNames_CB_Signif" = padjNames_CB_Signif)

######GLOBALS######
TotalPvalUnionCB <- character()
TotalPadjUnionCB <- character()

for (i in 1:2) {
  if (names(PCBlist[i]) == "pvaluesNames_CB_Signif") {
    for (position in 1:length(PCBlist[[i]])) {
      TotalPvalUnionCB <-
        base::unique(c(TotalPvalUnionCB,
pvaluesNames_CB_Signif[[position]]))
    }
  }
  else{
    for (position in 1:length(PCBlist[[i]])) {
      TotalPadjUnionCB <-
        base::unique(c(TotalPadjUnionCB,
padjNames_CB_Signif[[position]]))
    }
  }
```

```
}
length(TotalPvalUnionCB)#736
length(TotalPadjUnionCB)#248


#####Per Pesticide#########
PestPvalUnionCB <- character()
PestPadjUnionCB <- character()

for (i in 1:2) {
  if (names(PCBlist[i]) == "pvaluesNames_CB_Signif") {
    for (position in 1:4) {
      PestPvalUnionCB <-
        base::unique(c(PestPvalUnionCB,
pvaluesNames_CB_Signif[[position]]))
    }
  }
  else{
    for (position in 1:4) {
      PestPadjUnionCB <-
        base::unique(c(PestPadjUnionCB,
padjNames_CB_Signif[[position]]))
    }
  }

}
length(PestPvalUnionCB)#487
length(PestPadjUnionCB)#71


#####Por interaccion Pesticida y Sexo#########

InteraccPvalUnionCB <- character()
InteraccPadjUnionCB <- character()
names(pvaluesNames_CB_Signif)
for (i in 1:2) {
  if (names(PCBlist[i]) == "pvaluesNames_CB_Signif") {
    for (position in 5:8) {
      InteraccPvalUnionCB <-
        base::unique(c(InteraccPvalUnionCB,
pvaluesNames_CB_Signif[[position]]))
    }
  }
  else{
    for (position in 5:8) {
      InteraccPadjUnionCB <-
        base::unique(c(InteraccPadjUnionCB,
padjNames_CB_Signif[[position]]))
    }
  }

}
length(InteraccPvalUnionCB)#638
length(InteraccPadjUnionCB)#233


# HP ----------------------------------------------------------------
-----

#Expresion matrixes
protHP <-
```

```
  read.delim(
    "~/Dropbox/denamic/data/set03/02_Set03_HP_PostArsyn",
    header = T,
    row.names = 1,
    sep = " "
  )
#eliminacion CYP
protHP <- protHP[-which(rownames(protHP) == "CYPER_M_I6_HP"), ]

##voom must be used if we need to log transf the data

sex_hp <- factor(designHP[rownames(protHP), "Gender"])

pest_hp = factor(designHP[rownames(protHP), "Pesticide"], levels =
c("VH", "CHLOR01", "CHLOR03", "CHLOR1", "CAR"))


prot_matrix_HP = model.matrix( ~ sex_hp + pest_hp + sex_hp * pest_hp)

fit_HP = lmFit(t(protHP), prot_matrix_HP)

fit_p <- eBayes(fit_HP)


# TopRankedHP ------------------------------------------------------------
-----

betas <-
  colnames(fit_p$p.value)[-c(1, 2)] #quitamos intercept y la que solo
estudia el efecto del sexo ya que no nos interesan
TopRankedHPLists <- list()

for (element in betas) {
  TopRankedHPLists[[element]] <-
    topTable(fit_p, coef = element, number = 1223)
}


pvaluesNumber_HP_Signif <- list()
padjNumber_HP_Signif <- list()

pvaluesNames_HP_Signif <- list()
padjNames_HP_Signif <- list()

for (position in 1:length(TopRankedHPLists)) {
  pvaluesNames_HP_Signif[[names(TopRankedHPLists[position])]] <-
    character()

  padjNames_HP_Signif[[names(TopRankedHPLists[position])]] <-
    character()

  for (row in 1:nrow(TopRankedHPLists[[position]])) {
    if (TopRankedHPLists[[position]][row, 4] < 0.05) {
      pvaluesNames_HP_Signif[[names(TopRankedHPLists[position])]] <-
        c(pvaluesNames_HP_Signif[[names(TopRankedHPLists[position])]],
rownames(TopRankedHPLists[[position]][row, ]))
    }
  }

  pvaluesNumber_HP_Signif[[names(TopRankedHPLists[position])]] <-
```

```
length(pvaluesNames_HP_Signif[[names(TopRankedHPLists[position])]])

  for (row in 1:nrow(TopRankedHPLists[[position]])) {
    if (TopRankedHPLists[[position]][row, 5] < 0.05) {
      padjNames_HP_Signif[[names(TopRankedHPLists[position])]] <-
        c(padjNames_HP_Signif[[names(TopRankedHPLists[position])]],
rownames(TopRankedHPLists[[position]][row, ]))
    }
  }

  padjNumber_HP_Signif[[names(TopRankedHPLists[position])]] <-
    length(padjNames_HP_Signif[[names(TopRankedHPLists[position])]])
}

# Unions-Unique analysis ---------------------------------------------
------------
PHPlist <-
  list("pvaluesNames_HP_Signif" = pvaluesNames_HP_Signif,
       "padjNames_HP_Signif" = padjNames_HP_Signif)

######GLOBALS######
TotalPvalUnionHP <- character()
TotalPadjUnionHP <- character()

for (i in 1:2) {
  if (names(PHPlist[i]) == "pvaluesNames_HP_Signif") {
    for (position in 1:length(PHPlist[[i]])) {
      TotalPvalUnionHP <-
        base::unique(c(TotalPvalUnionHP,
pvaluesNames_HP_Signif[[position]]))
    }
  }
  else{
    for (position in 1:length(PHPlist[[i]])) {
      TotalPadjUnionHP <-
        base::unique(c(TotalPadjUnionHP,
padjNames_HP_Signif[[position]]))
    }
  }

}

#####Per Pesticide#########

PestPvalUnionHP <- character()
PestPadjUnionHP <- character()

for (i in 1:2) {
  if (names(PHPlist[i]) == "pvaluesNames_HP_Signif") {
    for (position in 1:4) {
      PestPvalUnionHP <-
        base::unique(c(PestPvalUnionHP,
pvaluesNames_HP_Signif[[position]]))
    }
  }
  else{
    for (position in 1:4) {
      PestPadjUnionHP <-
        base::unique(c(PestPadjUnionHP,
padjNames_HP_Signif[[position]]))
```

```
    }
  }

}
length(PestPvalUnionHP)#377
length(PestPadjUnionHP)#40


#####Por interaccion Pesticida y Sexo#########

InteraccPvalUnionHP <- character()
InteraccPadjUnionHP <- character()

for (i in 1:2) {
  if (names(PHPlist[i]) == "pvaluesNames_HP_Signif") {
    for (position in 5:8) {
      InteraccPvalUnionHP <-
        base::unique(c(InteraccPvalUnionHP,
pvaluesNames_HP_Signif[[position]]))
    }
  }
  else{
    for (position in 5:8) {
      InteraccPadjUnionHP <-
        base::unique(c(InteraccPadjUnionHP,
padjNames_HP_Signif[[position]]))
    }
  }

}
length(InteraccPvalUnionHP)#508
length(InteraccPadjUnionHP)#103


# todos los objetos a guardar ----------------------------------------
-----

save(
  cvCB_3,
  cvHP_3,
  padjNames_CB_Signif,
  TotalPadjUnionCB,
  PestPadjUnionCB,
  InteraccPadjUnionCB,
  padjNames_HP_Signif,
  TotalPadjUnionHP,
  PestPadjUnionHP,
  InteraccPadjUnionHP,
  file =
"~/Dropbox/denamic/data/set03/03_Set03_ProteinsSelectedBycv_3_ANDlimma
.RData"
)

# Union del total de proteinas de Expresion diferencial y de cv ------
-----

#CB
length(TotalPadjUnionCB)
length(cvCB_3)
TotalTotalUnionCB <- base::union(TotalPadjUnionCB, cvCB_3)
length(TotalTotalUnionCB)#658
```

```
#HP
length(TotalPadjUnionHP)
length(cvHP_3)
TotalTotalUnionHP <- base::union(TotalPadjUnionHP, cvHP_3)
length(TotalTotalUnionHP)#529

save(TotalTotalUnionCB, TotalTotalUnionHP, file =
"~/Dropbox/denamic/data/set03/03_Set03_TotalTotalUnionOfProteinsCommin
gFromLimmaAndCv.RData")
```