Universitat Oberta de Catalunya

uoc.edu

GoTerMinator

Pilar Cámara Castaño
Máster Bioinformática y Bioestadística
Área 2 Aula 1

Nombre Consultor/a Luis Franco Serrano
Nombre Profesor/a responsable de la asignatura Carles Ventura Royo

02 junio 2022

Salto de página

FICHA DEL TRABAJO FINAL

Título del trabajo:	*GoTerMinator*
Nombre del autor:	*Pilar Cámara Castaño*
Nombre del consultor/a:	*Luis Franco Serrano*
Nombre del PRA:	
Fecha de entrega :	*02/06/2022*
Titulación:	*Master en Bioestadística y Bioinformática*
Área del Trabajo Final:	
Idioma del trabajo:	*Español*
Número de créditos:	*15*
Palabras clave	*detector proteinas moonlighting*

Resumen de Trabajo

Hasta ahora las proteinas moonlighting han sido identificadas por métodos experimentales. Este trabajo analiza la posibilidad de crear una herramienta bioinformática capaz de indentificarlas.

Partiendo de la información de las bases de datos públicas Gene Ontology y Uniprot, se genera una matriz de distancias entre pares de términos GO, a partir de la cual se analiza la posibilidad de agrupar los términos GO según sus funcionalidades, y definir un espacio multidimensional que permita posicionar y medir la distancia entre términos GO. El objetivo final es determinar si las anotaciones GO de una proteína pertenecen a clusters diferentes, y/o si su posición en el espacio dimensional es distante, y por tanto la proteína puede ser moonlighting.

Como resultado de este análisis, se han obtenido una serie de agrupaciones de términos GO, mediante diferentes métodos, que se comparan sobre una base de datos de proteínas previamente identificadas como moonlighting, así como sobre proteínas de Uniprot de las que se desconoce su multifuncionalidad.
Es necesario analizar la especificidad y sensibilidad de los clusters obtenidos, para validar y optimizar los resultados. Esto requiere conocimientos biológicos que están fuera del alcance este trabajo.

En general, los estudios basados en Gene Ontology comparan las funciones biológicas de distintos genes o proteínas, para estudiar su similaridad. Lo novedoso de GOTerminator es que el objetivo final es encontrar la diferencia biológica dentro de un mismo gen o proteína

Abstract

Until now moonlighting proteins have been identified by experimental methods. This work analyzes the possibility of creating a bioinformatic tool capable of identifying them.

Based on the information from the public databases Gene Ontology and Uniprot, a matrix of distances between pairs of GO terms is generated, from which the possibility of grouping the GO terms according to their functionalities and defining a multidimensional space is analyzed. that allows positioning and measuring the distance between GO terms. The final objective is to determine if the GO annotations of a protein belong to different clusters, and/or if its position in the dimensional space is distant, and therefore the protein can be moonlighting.

As a result of this analysis, a series of groupings of GO terms have been obtained, by means of different methods, which are compared on a database of proteins previously identified as moonlighting, as well as on Uniprot proteins whose multifunctionality is unknown.
It is necessary to analyze the specificity and sensitivity of the clusters obtained, to validate and optimize the results. This requires biological knowledge that is outside the scope of this work.

In general, studies based on Gene Ontology compare the biological functions of different genes or proteins, to study their similarity. The novelty of GOTerminator is that the ultimate goal is to find the biological difference within the same gene or protein

ÍNDICE

LISTA DE FIGURAS

1. Introducción

1.1. Contexto y justificación del Trabajo

La temática escogida para este TFM es la identificación de proteínas Moonlighting o proteínas multifuncionales. Se trata de proteínas capaces de desempeñar dos o más funciones biológicas bien diferenciadas en el mismo organismo.[1] [3] [4]

Actualmente se sabe que:

- un 78% de las proteínas moonlighting humanas están relacionadas con enfermedades y la invasión de tejidos en cáncer.
- si una proteína moonlighting es objetivo de un fármaco, puede causar mayores efectos secundarios y toxicidad..
- un 25% de todas las proteínas multifuncionales identificadas hasta la fecha, están implicadas en los mecanismos de virulencia de los microorganismos patógenos,en gran parte en el mecanismo de adhesión al huésped.

Por tanto, identificar las proteínas moonlighting es de gran importancia para el análisis y la comprensión de las enfermedades humanas así como para el diseño de vacunas y fármacos dirigidos a objetivos.

La mayoría de proteínas moonlighting han sido identificadas por medios experimentales o por casualidad.

La búsqueda de proteínas moonlighting directamente de la bibliografía resulta difícil ya que la mayoría de autores desconocen el fenómeno de la multifuncionalidad de las proteínas, y no hacen referencia al término moonlighting en sus trabajos para referirse a este fenómeno, aunque en sus publicaciones estén realmente describiendo proteínas con características multifuncionales. (J. Cedano)

En este contexto, el desarrollo de herramientas bioinformáticas que ayuden a determinar qué proteínas pueden ser multifuncionales puede ser de gran utilidad, teniendo en cuenta la relevancia de estas proteínas en distintos campos de la biología, medicina y farmacología, así como el creciente número de genomas que se están secuenciando en la actualidad.

Existe gran cantidad de información acerca de las funciones moleculares y biológicas de las proteínas en bases de datos como Uniprot y ontológicas como Gene Ontology [7] [8] [9]

En este proyecto vamos a intentar encontrar proteínas multifuncionales aun no descritas como tales, mediante el desarrollo de una herramienta bioinformática capaz de identificar las proteínas que tienen asociados en Uniprot, anotaciones funcionales muy distantes.

1.2 Objetivos del Trabajo

Objetivo final

Desarrollar una herramienta informática que dado el identificador de una proteína, busque los términos GO de sus funciones en la base de datos Gene Ontology , y según los parámetros especificados, calcule la «distancia» entre ellos para determinar si corresponden a diferentes funciones biológicas, y de esta forma identificar si la proteína puede ser moonlighting.

Objetivos parciales

- Estudiar y analizar el concepto de «distancia» entre dos términos GO dados

- Desarrollar un algoritmo para calcular la «distancia» entre dos términos GO dados según distintos métodos : Distancia topológica ponderada (Tanimoto), distancia semántica y distancia por concurrencia en anotaciones en Uniref

- Mediante escalado multidimensional, establecer un sistema de coordenadas para posicionar los distintos términos GO

- Realizar una clusterización de los GO-terms, por criterio OBO, GOGO, concurrencia UniProt y usando la matriz de distancias conjunta.

- Desarrollar el detector de moons mediante distintas estrategias:

    - Estrategia distancia:
    - Estrategia de clustering
    - Estrategia mixta

- Probar si el programa diseñado es capaz de predecir las proteínas moonlighting de la base de datos MultitaskProtDB-II http://wallace.uab.es/multitaskII/proteins_list.php [2]

1.3. Enfoque y método seguido

Figura 1 - Obtener Datos

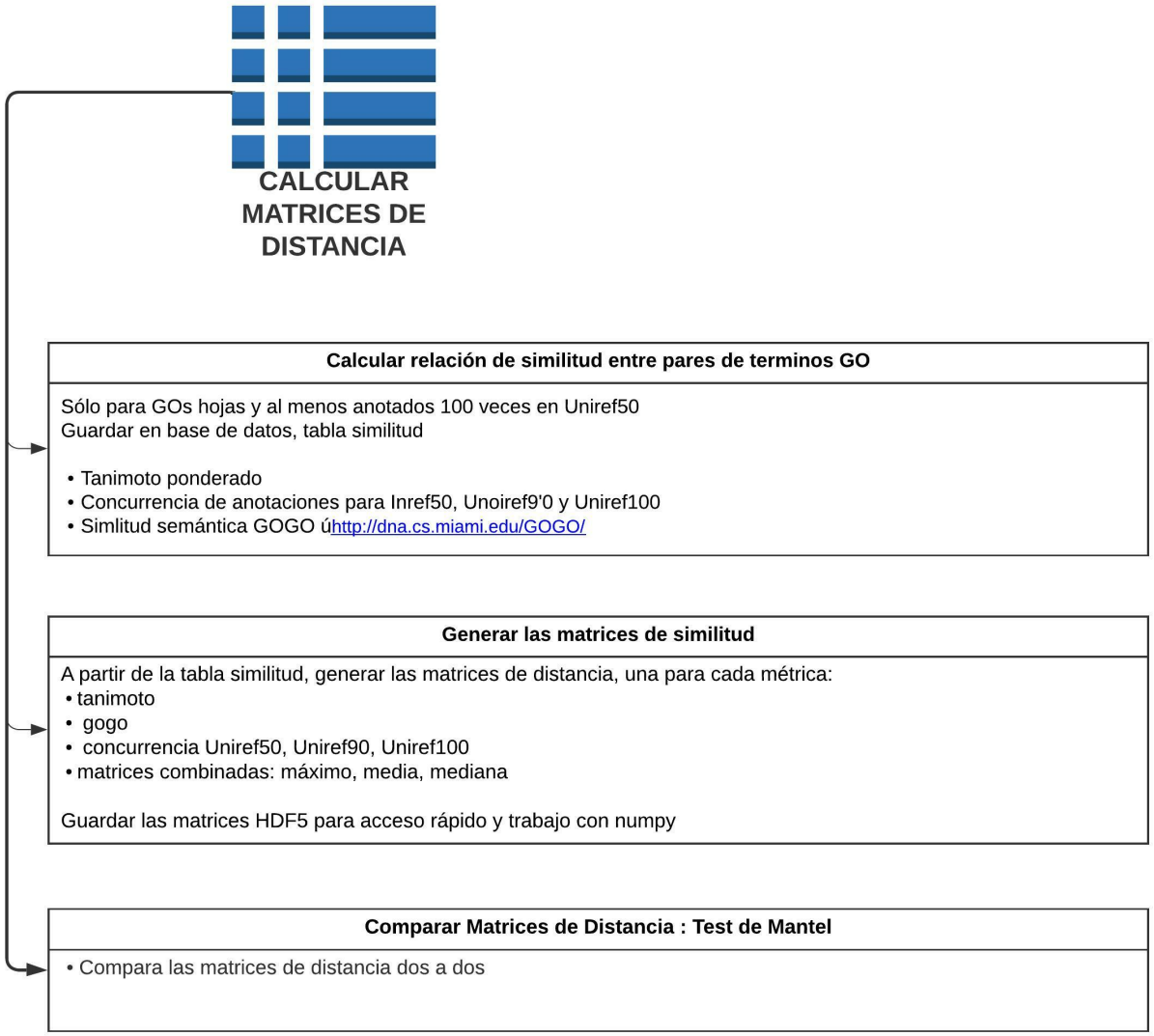


Figura 2 - Calcular Matrices de Distancia

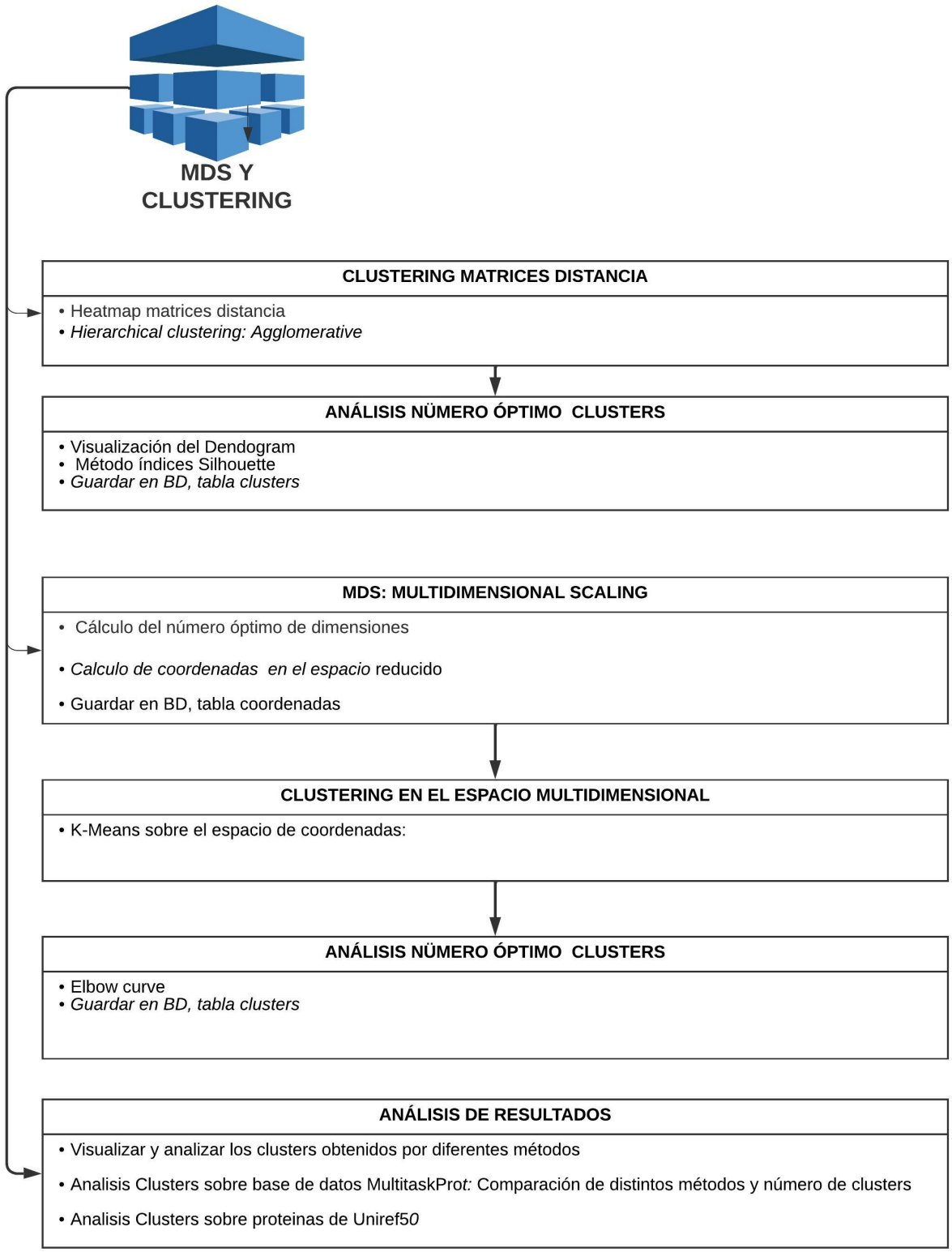


Figura 3 - MDS y Clustering

1.4. Planificación del Trabajo

EL trabajo se ha planificado teniendo en cuenta las fechas de entrega de las distintas PEC

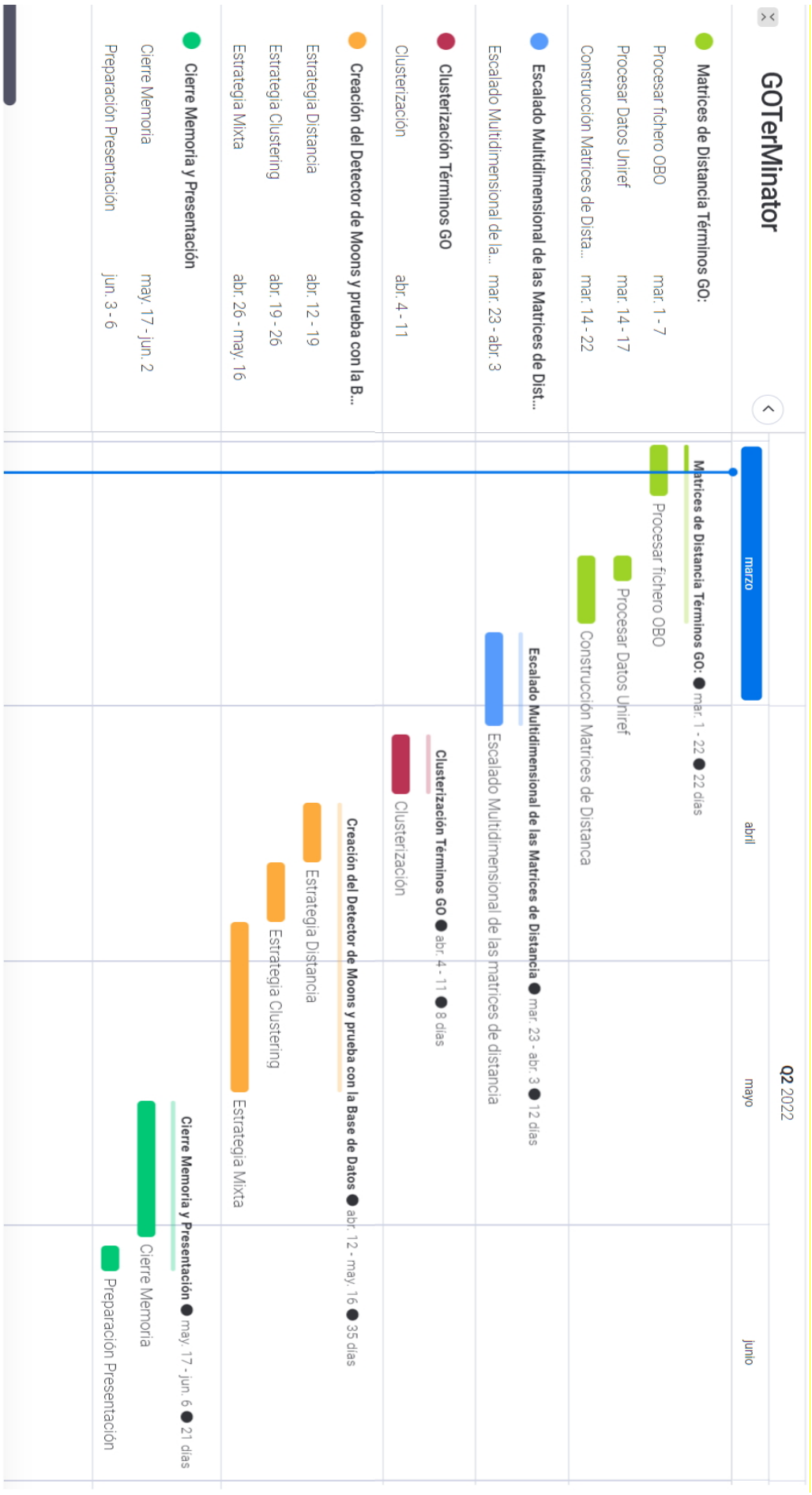


Figura 4 - Planificación

Hitos

Se corresponden con las entregas de las distintas partes del proyecto, sincronizadas como unidades funcionales con las fechas de entrega de las PEC

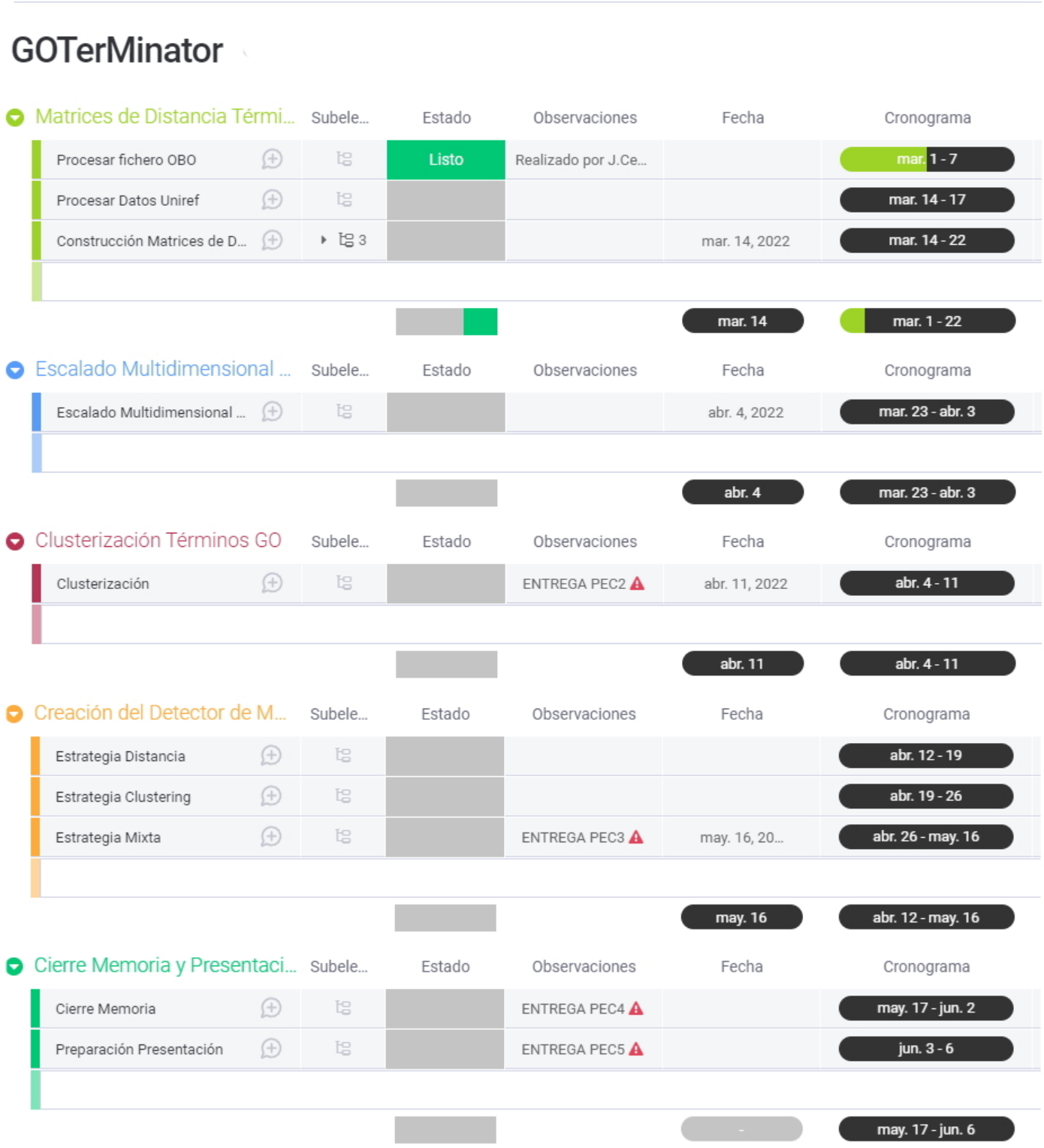


Figura 5 - Hitos

1.5. Breve sumario de contribuciones y productos obtenidos

- Estructura de base de datos MySQL para guardar datos de partida y resultados
- Notebook Jupyter: Programa Python que permite reproducir la obtención de datos y generación de matrices de distancia, pudiendo cambiar varias opciones mediante parámetros, entre ellos el tipo de ontologia (FUN, PRO, COM) y el subconjunto de Gos con los que se trabaja
- Notebook Jupyter: Programa Python que permite reproducir el análisis de clusters , pudiendo cambiar varias opciones mediante parámetros, así como ejecutarlo para cada tipo de ontología (FUN, PRO, COM)
- Tabla MySQL «similitud», con la distancia por pares del subconjunto de términos GOs seleccionado
- Matrices de distancia en formato HDF5 para distintas métricas: Tanimoto ponderado, semántica GOGO, concurrencia anotaciones en Uniref, combinadas por máximo, media y mediana
- Listado Clusters obtenidos mediante diferentes métodos
- Coordenadas de los GOs en el espacio multidimensional
- Listado de proteínas moonlighting de MultitaskProtdb con sus anotaciones GO y descripción, número de hijos en el DAG, y número de anotaciones en Uniref
- Listado de proteínas de Multitaskprot con sus anotaciones de GO (solo del subconjunto elegido) y comparación de distintos clusters
- Listado de proteínas de proteínas de Uniprot (seleccionadas de forma aleatoria), que se desconoce si son moonlighting, con sus anotaciones GO (solo del subconjunto elegido) y comparación de distintos clusters.

2. Estado del Arte

La búsqueda de proteínas Moonlighting hasta la fecha se viene realizando de forma experimental. Identificar proteínas moonlighting directamente de la bibliografía se ve altamente dificultado por el hecho de que la mayoría de los autores desconocen el fenómeno de la multifuncionalidad de las proteínas, y hacen referencia al término moonlighting o Multifuncional en sus publicaciones, aunque claramente estén describiendo proteínas con características multifuncionales.

La información de Gene Ontology se utiliza actualmente en múltiples aplicaciones como: estudios de genómica funcional (análisis de enriquecimiento), categorización funcional, predicción de funciones de productos génicos, minería de datos , análisis semánticos.

En particular, la similitud semántica se ha utilizado en aplicaciones como:

- comparación genes, predicción de funciones y evaluación y validación de métodos automáticos de predicción
- en el análisis de datos de transcriptómica y proteómica, permite mejorar la agrupación de genes expresados teniendo en cuenta su similitud funcional, comparar resultados de experimentos diferentes, mejorar la calidad de los datos y validar la selección de genes con fines biomédicos. Otras aplicaciones incluyen
- evaluar el significado biológico de dominios cromosómicos co-expresados
- predicción de la localización celular

[28]

En general Gene Ontology se ha utilizado para comparación de similitud entre funciones biológicas de distintos productos génicos. El objetivo de GOTerMinator es encontrar la diferencia biológica entre distintas anotaciones GO de un mismo gen, para identificar proteínas moonlighting.

Por otro lado, la distancia semántica entre términos de Gene Ontology y su clusterización, aunque existen varios algoritmos y se han realizado algunos análisis, es aún un tema de investigación..

Por tanto GOTerMinator supone:

- una nueva aplicación del análisis semántico de GO
- una nueva herramienta bioinformática para ayudar a identificar proteínas multifuncionales

3. Metodología

A continuación se presenta una descripción detallada de la metodología seguida en cada paso:

INFORMACIÓN DE BASES DE DATOS DE LIBRE ACCESO

- UNIPROT (Universal Protein) : contiene la secuencia de proteínas, información de funciones e índice de artículos de investigación. Integra recursos de tres bases de datos principales, incluidas EBI (Instituto Europeo de Bioinformática), SIB (Instituto Suizo de Bioinformática) y PIR (Recurso de información de proteínas). Actualmente, UniProt se compone de diversas sub-bibliotecas

    - UnitProtkB/Swiss-Prot : Base de datos no redundante de alta calidad anotada a mano
    - UniprotKB/TrEMBL : Traducción automática de secuencias de proteínas, secuencias predichas, bases de datos no verificadas
    - UniParc : Base de datos de secuencias de proteínas no redundantes
    - UniRef: (Clústeres de referencia de UniProt): base de datos de secuencias no redundantes. Las secuencias son agrupadas por diversos niveles de identidad de secuencia. Existe en tres versiones: UniRef100, UniRef90, y UniRef50, que agrupan secuencias con 100 %, ≥90 %, y ≥50 % de identidad, respectivamente.
    - Proteomes: Proporciona información proteómica para especies de genoma completamente secuenciadas

En este proyecto se utiliza UnioritKb/Swiss-Prot y Uniref50, Uniref90 y Uniref100
[10] [11]

- GENE ONTOLOGY28 may 2022 : La ontología GO es una herramienta bioinformática cuyo objetivo es estandarizar la representación de los genes y los atributos de sus productos génicos de todas las especies. Para ello organiza los términos GO, que representan distintos atributos en un DAG en el cual los términos son vértices o nodos y las relaciones entre ellos son los arcos. GO se divide en 3 ontologías (jerarquías):

    - Proceso biológico: (PRO) Eventos celulares a los que contribuye el producto génico.
    - Función molecular (FUN): Descripción bioquímica del producto génico.
    - Componente celular (COM) Localización o complejos de los que forma parte el gene o su producto génico.

Los tipos de arcos en GO representan la relación entre los distintos términos: «is a», «part of», »regulates»
[9]
Gene Ontology se puede descargar en diversos formatos: OBO, OWL, JSON. Para este proyecto utilizaremos el fichero go-basic.obo. Es la versión báscia de GO en formato texto, filtrada de forma que se garantiza que el gráfico es acíclico y las anotaciones se pueden propagar hacia arriba en el gráfico. Además se excluyen las relaciones entre las 3 jerarquías GO

[8] [12]

- MultitaskProtDB

Es un repositorio de proteínas multifuncionales encontradas en la literatura que contiene información relevante para cada una de las entradas, como números de acceso NCBI, EC y UniProt, funciones biológicas canónicas y moonlighting , estados monoméricos/oligoméricos, códigos PDB cuando están disponibles y referencia bibliográfica. Esta base de datos se utilizará para probar si la herramienta desarrollada es capaz de identificar estas proteínas

DEFINICIÓN DE SIMILITUD Y DISTANCIA

- Similitud

Definición *Una medida de similitud entre dos objetos P y Q, $s(P,Q)$, debe cumplir la siguientes propiedades:*

(I) Simetría: $s(P,Q) = s(Q,P)$

(II) No negativa: $s(P,Q) \geq 0$

(III) Identificación del objeto: $s(P,Q) \leq s(P,P)$

[29]

- Distancia: D = 1- Similitud
  Debe cumplir las siguientes propiedades

1. $d(x,y) = 0 \Leftrightarrow x = y$
2. $d(x,y) = d(y,x)$ (simetría)
3. $d(x,z) \leq d(x,y) + d(y,z)$ (desigualdad triangular).

De estos también se deduce:

> $d(x,y) \geq 0$ (no negatividad)

[30]

Una distancia es semimétrica o seudométrica si cumple sólo las tres primeras propiedades.

MEDIDAS DE SIMILITUD UTILIZADAS ENTRE TÉRMINOS DE GENE ONTOLOGY

Los términos en una ontología con estructura de grafo, como GO, pueden compararse mediante dos métodos fundamentales:

- Los métodos basados en aristas: consisten en contar el número de arcos entre dos nodos del grafo.. Se basan en dos supuestos que son ciertos en raras ocasiones en la biología: (1) los nodos y las aristas están distribuidos uniformemente y (2) las aristas en el mismo nivel de la ontología se corresponden con igual distancia semántica entre los términos. Para compensar estos problemas pueden utilizarse pesos .

En este proyecto se utiliza el índice de Jaccard (I_J), ponderado, que define la similitud entre dos términos A y B como .

$$\mathcal{J}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

- Los métodos basados en nodos: utilizan las propiedades de los términos implicados, que pueden relacionarse con los propios términos, sus ancestros o descendientes. Uno de los conceptos empleados es el Contenido de Información (IC), que se calcula en base a la ocurrencia de un término en una base de datos (como Uniprot), o a partir del número de hijos que tiene un término en GO. Son métodos menos sensibles a la variabilidad de distancia semántica y densidad de nodos, porque el IC es independiente de su profundidad en la ontología. Pero puede estar sesgado por las tendencias actuales de la investigación biomédica, pues los términos de interés científico tienen más probabilidad de estar anotados. Además el cálculo es bastante costoso porque requiere grandes bases de datos de anotaciones
  [13] [14] [15] [16] [17] [18] [28]

En este proyecto se utiliza la librería GOGO que es un algoritmo híbrido mejorado que imita la propiedad de IC sin calcular IC

[32]

- Concurrencia anotaciones: se basa en la ocurrencia simultánea de términos en el mismo conjunto de entradas de Uniprot. La coincidencia de términos GO revela vínculos biológicos naturales entre las funciones GO [19]

En este proyecto se calcula la Concurrencia entre dos términos A y B como

$$C(A,B)= \frac{\text{Frecuencia}(A \cap B)}{\text{Frecuencia}(A \cap B)}$$

MATRICES DE SIMILITUD Y DISTANCIA*:* Son matrices simétricas que guardan la medida de similitud entre cada par de términos GO. SE han generado 3 matrices de distancia, una para cada métrica , así como 3 matrices combinadas que se calculan como el máximo, media y mediana (elemento a elemento) de las 3 matrices

Las matrices obtenidas están normalizadas, todos sus valores están en el intervalo [0,1]

Existen diversas formas de calcular la matriz de distancia a partir de una matriz de similitud. SE ha utilizado Distancia = 1- Similitud

TEST DE MANTEL : COMPARACIÓN DE LAS MATRICES DE DISTANCIA

El test de Mantel estima el grado de correlación existente entre dos matrices X e Y. La hipótesis nula de esta técnica (Ho) es que las distancias/similitudes entre las variables de la matriz Y no están linealmente correlacionados con las correspondientes distancias/similitudes en la matriz X.
De este modo podemos evaluar si las diferentes medidas de distancia entre términos GOs están correlacionadas y por tanto son equivalentes y en ese caso utilizar una medida de distancia combinada, que aportaría la información de la relación entre cada par de términos GOs considerando los diferentes aspectos: distancia en el grafo, distancia semántica, cocurrencia en las anotaciones de Uniref
El estadístico del test de Mantel (ZM) se calcula mediante la suma de los productos cruzados de los valores de las dos matrices de similitud/distancia, excluyendo la diagonal principal que sólo contiene valores triviales (0 en el caso de las matrices de distancias y 1 en el caso de las matrices de similitudes).
$ZM = \Sigma X_{ij} Y_{ij}$
Donde Xij e Yij son los elementos de las matrices X e Y, respectivamente. Para evaluar la significación se realiza un test de permutaciones, en el que los elementos de una matriz se reordenan al azar y se calcula iterativamente el valor de Z. De la distribución de valores Z obtenidos al azar podemos evaluar cuál es la probabilidad de obtener el valor Z observado.

La correlación puede computarse por dos métodos::

- el coeficiente de Correlación del momento del producto de Pearson: Evalúa la relación lineal entre dos variables continuas. Una relación es lineal cuando un cambio en una variable se asocia con un cambio proporcional en la otra variable.

- el coeficiente de correlación del orden de los rangos de Spearman: evalúa la relación monótona entre dos variables continuas u ordinales. En una relación monótona, las variables tienden a cambiar al mismo tiempo, pero no necesariamente a un ritmo constante. El coeficiente de correlación de Spearman se basa en los valores jerarquizados de cada variable y no en los datos sin procesar.

HEATMAPS

Representan una matriz de valores mostrando, en lugar de números, un gradiente de color proporcional al valor de cada variable en cada posición. La combinación de un dendrograma con un heatmap permite ordenar por semejanza las filas y o columnas de la matriz, a la vez que se muestra con un código de colores el valor de las variables. Facilita la identificación visual de posibles patrones característicos de cada cluster.
[33]

ESPACIO MULTIDIMENSIONAL A PARTIR DE UNA MATRIZ DE DISTANCIAS

El escalado Multidimensional (MDS) es una técnica de análisis multivariante que, partiendo de una matriz de distancias entre individuos, crea una representación de los individuos en una escala euclidea ordinaria de unas cuantas variables (menor que la dimensión de la matriz), de modo que las distancias en dicha escala representen a las distancias de partida.
[34]

Es importante elegir el número de dimensiones adecuadas, de forma que se reduzca la dimensionalidad pero sin perder información relevante. Esto se mide mediante el concepto del stress

Kruskal (1964) sugiere las siguientes interpretaciones del Stress:

- 0.2 → Pobre
- 0.1 → Aceptable
- 0.05 → Bueno
- 0.025 → Aceptable
- 0.0 → Excelente

Una forma de hacerlo es ejecutar MDS para diferentes dimensiones y trazar la curva de stress para cada valor. Como el stress disminuye al aumentar las dimensiones, la clave es encontrar un número de dimensiones que mantenga un equilibrio con el stress. Normalmente, al principio el stress disminuye mucho al aumentar el número de dimensiones, y posteriormente la curva se estabiliza. Por ello, suele elegirse el codo (elbow) de la curva como valor óptimo para el número de dimensiones
[20] [21] [22] [23][35]

CLUSTERING

Una vez que se tiene el espacio de coordenadas se trata de ver que elementos están mas próximos o más distantes. También es posible realizar el clustering directamente a partir de las matrices de distancia.

El término clustering hace referencia a diversas técnicas cuyo objetivo es encontrar patrones o grupos (clusters) dentro de un conjunto de observaciones, de forma que las observaciones que están dentro de un mismo grupo son similares entre ellas y distintas a las observaciones de otros grupos. Es un método de aprendizaje no supervisado, porque no se tiene en cuenta (o no se conoce) a qué grupo pertenece realmente cada observación, a diferencia de los métodos de clasificación supervisada. en los que sí se emplea la verdadera clasificación durante su entrenamiento.

Existen tres tipos principales de clustering:

- Partitioning Clustering: requieren que el usuario especifique de antemano el número de clusters que se van a crear: Por ejemplo, K-means, K-medoids

- Hierarchical Clustering: no requieren que el usuario especifique de antemano el número de clusters. por ejemplo, agglomerative clustering, divisive clustering.

- Métodos combinados: hierarchical K-means, fuzzy clustering, model based clustering y density based clustering

En este proyecto se han utilizado los métodos Agglomerative Cluster (sobre las matrices de distancia) y K-Means (sobre el espacio de coordenadas)

AGGLOMERATIVE CLUSTERING

EL algoritmo es el siguiente

- Considerar cada una de los n individuos como un cluster individual, formando así la base del dendrograma (hojas).
- Calcular la distancia entre cada posible par de los n clusters, según el tipo de medida y de linkage elegido. El linkage cuantifica la distancia entre dos clusters. Existen diversos tipos de linkage:
    - Complete or Maximum: se calcula la distancia entre todos los posibles pares formados por una observación del cluster A y una del cluster B. y se elige la mayor de ellas como la distancia entre los clusters A y B
    - Single or Minimum: se calcula la distancia entre todos los posibles pares formados por una observación del cluster A y una del cluster B y se elige la menor de ellas como distancia entre A y B.
    - Average: Se calcula la distancia entre todos los posibles pares formados por una observación del cluster A y una del cluster B. El valor promedio de todas ellas se selecciona como la distancia entre los dos clusters (mean intercluster dissimilarity).
    - Centroid: Se calcula el centroide de cada uno de los clusters y se selecciona la distancia entre ellos como la distancia entre los dos clusters.
    - Ward: La selección del par de clusters que se combinan en cada paso se basa en el valor óptimo de una función objetivo, pudiendo ser esta última cualquier función definida por el analista., por ejemplo minimizar la suma total de varianza intra-cluster.
- Se fusionan los dos clusters mas similares, de forma que quedan n-1 clusters.
- Se repite el proceso hasta obtener el número de clusters deseado

K-MEANS

Es el algoritmo de clustering más utilizado. En Python no es posible utilizarlo directamente sobre matrices de distancia, solo sobre puntos en un espacio de coordenadas.K-Means tiene muy buena escalabilidad con la cantidad de datos. Para utilizar K-Means debemos especificar el número de grupos que queremos encontrar. A este número de grupos se le denomina K.

El algoritmo K-Means sigue los siguientes pasos:

- Inicialización: se elige la localización de los centroides de los K grupos aleatoriamente
- Asignación: se asigna cada dato al centroide más cercano
- Actualización: se actualiza la posición del centroide a la media aritmética de las posiciones de los datos asignados al grupo Los pasos 2 y 3 se siguen iterativamente hasta que no haya más cambios.

ESTIMACIÓN DEL NÚMERO DE CLUSTERS

Lo más complicado es determinar el número óptimo de clusters, especialmente en los algoritmos que requieren que se especifique para poder ver los resultados. Es más bien un proceso subjetivo que depende del algoritmo utilizado, y de la información previa que se tenga sobre los datos.

Existen algunos métodos que ayudan a estimar el número óptimo de clusters:

- Método Elbow

Consiste en probar un rango de valores para el número de clusters, representar gráficamente los resultados obtenidos con cada uno, e identificar el codo (elbow) o punto de la curva a partir del cual la mejora deja de ser notable. Calcula la varianza total intra-cluster en función del número de clusters y asigna como óptimo número de clusters el valor a partir del cual añadir más clusters apenas consigue mejoría.

- Método average silhouette

El coeficiente silhouette mide cómo de buena es la asignación de una observación a un cluster, comparando su similitud con el resto de observaciones de ese mismo cluster, frente a la similitud con las observaciones de los demás clusters. Su valor esta en el intervalo [-1, 1]. Los valores próximos a 1 indican que la observación se ha asignado al cluster correcto.

- Dendograma

Es un tipo de representación gráfica en forma de árbol que organiza y agrupa los datos en subcategorías según su similitud; Los objetos similares se unen por un enlace cuya posición está determinada por el nivel de similitud entre los objetos o grupos de objetos. Son muy útiles para visualizar gráficamente los clusters. Dependiendo a la distancia desde la raíz a la que se corte, se obtendrá un número diferente de clusters

Limitaciones del clustering

El clustering es muy útil para encontrar agrupaciones de datos, especialmente para grandes volúmenes de datos. Algunas de sus limitaciones a la hora de aplicarlo son:

- las decisiones que se tomen en relación a los parámetros (método, linkage, número de repeticiones, número de clusters, etc) influyen en gran medida en los resultados obtenidos y pueden tener grandes consecuencias. Como no suele haber una única respuesta correcta,deben probarse diferentes opciones. Esto es especialmente complicado cuando el conjunto de datos es muy grande
- Escalado y centrado de las variables
- Seleccionar la medida de distancia/similitud adecuada
- A qué altura establecer el corte de un dendrograma
- La validación de los clusters obtenidos, puede resultar difícil si se desconoce a priori la verdadera agrupación de los datos .
- Falta de robustez: los métodos de K-means-clustering y hierarchical clustering asignan obligatoriamente cada observación a un grupo. Si existe en la muestra algún outlier, a pesar de que realmente no pertenezca a ningún grupo, el algoritmo lo asignará a uno de ellos provocando una distorsión significativa del cluster en cuestión. Algunas alternativas son k-medoids y DBSCAN, aunque en este caso el número de outliers puede ser demasiado grande, especialmente cuando el volumen de datos es alto
- En hierarchical clustering si se realiza una mala división en los pasos iniciales, no se puede corregir en los siguientes pasos

[24]

LENGUAJE DE PROGRAMACIÓN

Se ha utilizado el lenguaje de programación Python. Entre las librerías utilizadas son relevantes:

- mysql.connector: permite el acceso a bases de datos Mysql
- GOATools, pygosemsim y mega-go permiten recorrer y extraer información de GO, así como calcular algunas de las distancias semánticas comentadas anteriormente.
- numpy para el trabajo con matrices y arrays
- mantel, scikit-learn, SciPy para los cálculos y análisis estadísticos
- h5py para trabajar con ficheros HDF5

BASE DE DATOS

Para guardar resultados intermedios y finales se ha utilizado una base de datos Mysql
Para guardar las matrices de distancia se ha utilizado un sistema de archivos HDF5

[25] [26] [27]

4. Resultados

La programación se ha hecho en Python y Mysql, mediante dos notebooks Jupyter, donde además del código pueden visualizarse los resultados a medida que van obteniendo. .

Los notebooks están parametrizados, de forma que es posible repetir todos los procesos, tanto de importación y preparación de datos como de análisis.

EL parámetro ontologia permite ejecutar el código para cada una de las ontologías, FUN, PRO, COM de Gene Ontology. Los parámetros Max_numero Anotaciones y Max_num_hijos, permiten obtener la lista de términos GO con la que se quiere trabajar. Otros parámetros permiten repetir solo parte de los procesos, partiendo de datos obtenidos y guardados en la base de datos o en HDF5 anteriormente

Como el volumen de datos es muy grande, ha sido necesario utilizar diccionarios Python y optimizar al máximo la programación, especialmente en los cálculos de distancias y los accesos cruzados a tablas Mysql

- GOTerMinator_MATRICES_DISTANCIA.ipynb
  - crear la estructura de la base de datos MoonDB
  - obtener los datos de Gene Ontology, Uniprot y MultitaskprotDB
  - Para cada tipo de ontologia (FUN, PRO, COM):
    - calcular la similitud entre cada par de términos GO, mediante las tres métricas: Tanimoto ponderado, gogo, Concurrencia Uniref
    - generar las matrices de similitud para cada métrica y las matrices de distancia combinadas
    - guardar las matrices de similitud en formato HDF5

- GOTerMinator_ANALISIS.ipynb
  - Para cada tipo de ontologia (FUN, PRO, COM):
    - Heatmaps
    - Agglomerative Clustering
    - Dendograms
    - Cácular número óptimo de clusters
    - Generar y guardar clusters en la base de datos
    - MDS: calcular el número óptimo de dimensiones del espacio de coordenadas
    - Calcular las coordenadas de cada término GO
    - Clustering sobre el espacio de coordenadas con K-Means
    - Visualización, comparación de los clusters obtenidos con los distintos métodos
    - Análisis de resultados (clusters):
      - ⇒ Sobre proteínas de Multitaskprotdb que sabemos que son moonlighting
      - ⇒ Sobre proteínas aleatorias de Uniref50, que desconocemos si son moonlighting

A continuación se explican los distintos procesos realizados resultados obtenidos

4.1. Obtener y Preparar los Datos

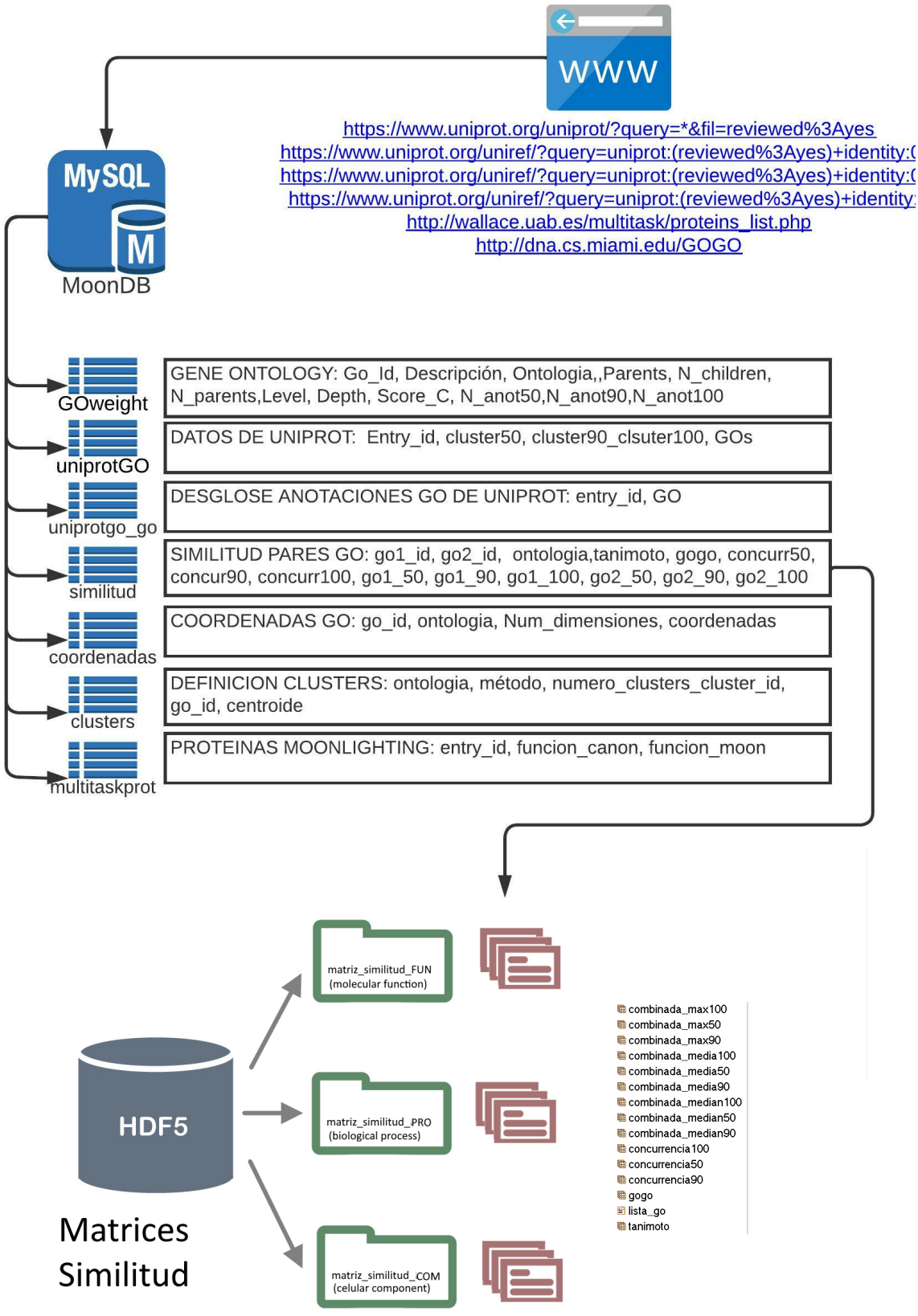


Figura 6 - Esquema Generar Matrices Similitud

Crear base de datos MoonDB y tablas (MySQL)

Se genera la base de datos MoonDB en un servidor local MySQL 5.7, y se crea la estructura de las tablas donde se van a guardar los datos de partida, algunos resultados intermedios, y los resultados finales obtenidos.

Obtener Datos

- ***Gene Ontology***
  - Descarga del fichero go_basic.obo desde http://geneontology.org/docs/download-ontology/
  - Leer el DAG con las funciones de la librería goatools y generar la tabla goweight con la relación de términos GO, su descripción, el número de hijos y de padres, y el peso (C_score) que posteriormente se utiliza para calcular la métrica Tanimoto
  - El peso (weight) es un número entero, que para un termino GO dado, se calcula mediante la siguiente función:
    peso(GO) = score_C= round(5*num_padres(GO)/(num_hijos(GO)+num_padres(GO)),0)+1
    Para la raíz será 0 y para las hojas será siempre 6. En estudios previos se ha probado que con estos valores se obtienen buenos resultados

- **Uniprot GO y Uniref50, Uniref90, Uniref1*00***

Descargar en formato CSV:

  - uniprot con anotaciones https://www.uniprot.org/uniprot/?query=*&fil=reviewed%3Ayes
  - Uniref100 https://www.uniprot.org/uniref/?query=uniprot:(reviewed%3Ayes)+identity:1
  - Uniref90 https://www.uniprot.org/uniref/?query=uniprot:(reviewed%3Ayes)+identity:0.9
  - Uniref50 https://www.uniprot.org/uniref/?query=uniprot:(reviewed%3Ayes)+identity:0.5

- Leer y cargar los datos en la base de datos, en las tablas uniprotgo, uniprotgo_go. Marcar en uniprotgo, las proteínas que pertenecen a Uniref50, Uniref90 y/o Uniref100

- A partir de los datos anteriores, calcular el número de anotaciones de cada término GO en Uniref50, Uniref90, Uniref100

MultitaskProtDB

- Descargar la tabla MultitaskProtDB desde http://wallace.uab.es/multitask/proteins_list.php en fromato csv
- Leer el fichero csv y guardar la información en la base de datos en la tabla "multitaskprot": id de Uniprot, la descripción de la función canónica y la descripción de la función moonlighting

La información de esta base de datos se utiliza

- en primer lugar para analizar el tipo de términos GOs que caracterizan a las funciones canónicas y moonlighting de estas proteínas
- y posteriormente para probar si los clusters que obtengo son capaces de identificar las diferentes funcionalidades de estas proteínas

Generar Matrices de Similitud

- lista_go_100: Existen 43786, términos GO en el fichero obo descargado. Ante la imposibilidad de trabajar con todos ellos, se ha seleccionado un subconjunto. Para este análisis he utilizado los parámetros min_anot = 100 y max_N_Children = 0, es decir los términos GO referenciados mas de min_anot veces en Uniref100 y con no mas de max_N_Children hijos. Los valores de min_anot y max_N_Children están parametrizados, de forma que es posible probar con diferentes valores para seleccionar la lista de términos GOs con los se quiere trabajar.

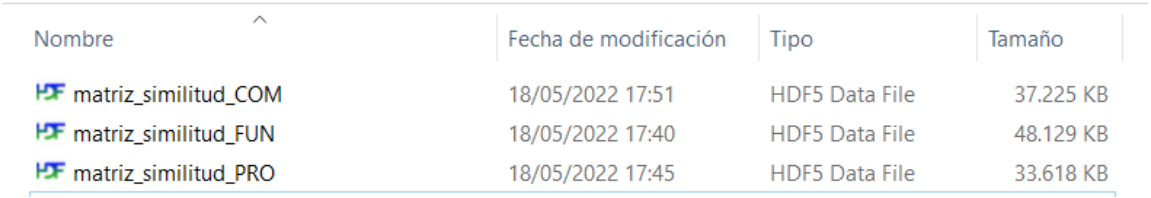
- Se calcula la similitud entre cada par de términos GO mediante las medidas explicadas anteriormente
    - Tanimoto ponderado según el score_c calculado
    - Semántica GOGO : para agilizar los cálculos, en lugar de programar una funciona propia o utilizar las librerías perl, debido al tiempo limitadi del proyecto, he optado por utilizar directamente la herramienta online http://dna.cs.miami.edu/GOGO . Para ello se generan varios ficheros con los pares de términos GO, cuya distancia se quiere calcular (hasta 100000 pares por fichero), se suben a la web, y posteriormente se leen los ficheros csv de resultados obtenidos por email
    - Concurrencia en las anotaciones: se calculan las frecuencias de cada par de términos, en Uniref100, Uniref90 y Uniref50, así como las frecuencias de cada término por separado.

- Se guardan en la base de datos, tabla similitud, los valores obtenidos para da par de términos GO: Tanimoto, gogo, frecuencia de la intersección y frecuencia por separado de cada uno de los dos términos

- A partir de la tabla similitud se crean las matrices de similitud según cada una de las métricas, así como 3 matrices combinadas:
    - Tanimoto
    - gogo
    - Concurrencia (50, 90, 100) frecuencia de la intersección/frecuencia de la unión
    - combinada máximo(Tanimoto,gogo, Concurrencia)
    - combinada media, solo de los valores > 0 (Tanimoto,gogo, Concurrencia)
    - combinada mediana solo de valores > 0 (Tanimoto,gogo, Concurrencia)

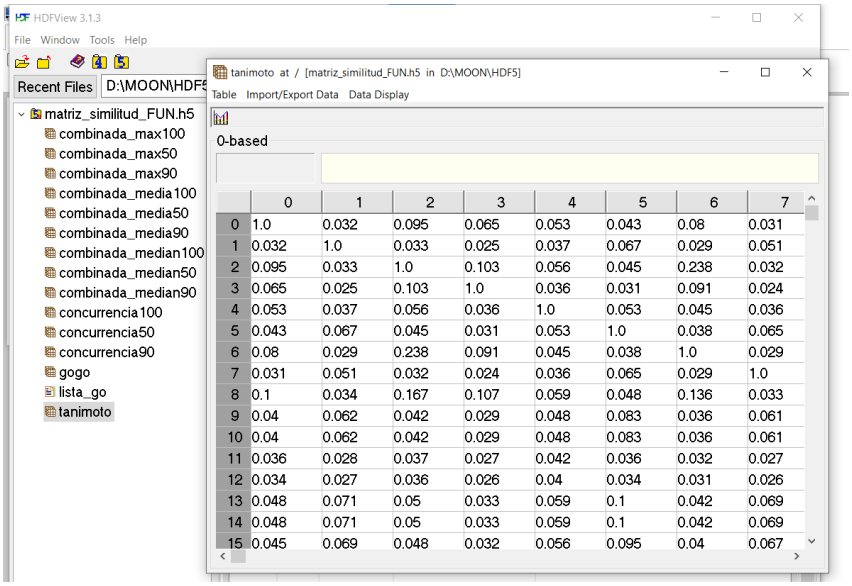
Todos los valores de similitud están en el intervalo [0, 1], 1 significa que los dos términos GO son iguales y 0 que no hay similitud entre ellos

- Se guardan las matrices de similitud y las lista de GOs (labels) en un archivo HDF5. Los archivos HDF5 permiten generar arrays numpy directamente y de forma inmediata, no teniendo que calcular las matrices cada vez que se necesiten



Nombre	Fecha de modificación	Tipo	Tamaño
matriz_similitud_COM	18/05/2022 17:51	HDF5 Data File	37.225 KB
matriz_similitud_FUN	18/05/2022 17:40	HDF5 Data File	48.129 KB
matriz_similitud_PRO	18/05/2022 17:45	HDF5 Data File	33.618 KB

Cada archivo HDF5 corresponde a un tipo de ontología y dentro de cada archivo un fichero para cada matriz



	0	1	2	3	4	5	6	7
0	1.0	0.032	0.095	0.065	0.053	0.043	0.08	0.031
1	0.032	1.0	0.033	0.025	0.037	0.067	0.029	0.051
2	0.095	0.033	1.0	0.103	0.056	0.045	0.238	0.032
3	0.065	0.025	0.103	1.0	0.036	0.031	0.091	0.024
4	0.053	0.037	0.056	0.036	1.0	0.053	0.045	0.036
5	0.043	0.067	0.045	0.031	0.053	1.0	0.038	0.065
6	0.08	0.029	0.238	0.091	0.045	0.038	1.0	0.029
7	0.031	0.051	0.032	0.024	0.036	0.065	0.029	1.0
8	0.1	0.034	0.167	0.107	0.059	0.048	0.136	0.033
9	0.04	0.062	0.042	0.029	0.048	0.083	0.036	0.061
10	0.04	0.062	0.042	0.029	0.048	0.083	0.036	0.061
11	0.036	0.028	0.037	0.027	0.042	0.036	0.032	0.027
12	0.034	0.027	0.036	0.026	0.04	0.034	0.031	0.026
13	0.048	0.071	0.05	0.033	0.059	0.1	0.042	0.069
14	0.048	0.071	0.05	0.033	0.059	0.1	0.042	0.069
15	0.045	0.069	0.048	0.032	0.056	0.095	0.04	0.067

Figura 7 - Matrices Similitud HDF5

- A partir de las matrices de similitud se generan las matrices de distancia, con las que se realiza el análisis matriz_distancia = 1- matriz de similitud

Análisis de Datos de Gene Ontology y MultitaskProtDB

- Gene Ontology: Número total de GOs de cada tipo

Tipo	COUNT(*)
COM	4.183
FUN	11.175
PRO	28.428

- Gene Ontology: Número de GOs de cada tipo con los que se han generado las matrices de distancia

	Tipo	COUNT(*)	
0	COM	583	min_anot= 50 , max_N_children=2
1	FUN	663	min_anot= 100 max_N_children=0
2	PRO	554	min_anot= 100 max_N_children=0

- Descripciones duplicadas en Gene Ontology

	GO	Description	Tipo	N_Children	N_annot_100
0	GO:0080043	quercetin 3-O-glucosyltransferase activity	FUN	0	68
1	GO:0080045	quercetin 3-O-glucosyltransferase activity	FUN	0	3
2	GO:0030755	quercetin 3-O-methyltransferase activity	FUN	0	8
3	GO:0102822	quercetin 3-O-methyltransferase activity	FUN	0	4
4	GO:0102446	rhamnetin 3-O-methyltransferase activity	FUN	0	0
5	GO:0102447	rhamnetin 3-O-methyltransferase activity	FUN	0	0
6	GO:0004800	thyroxine 5-deiodinase activity	FUN	0	33
7	GO:0033798	thyroxine 5-deiodinase activity	FUN	0	14

Los términos GO duplicados no están incluidos en el subconjunto con el que se han generado las matrices de distancia, puesto que todas tienen menos de 100 anotaciones en Uniref100

- MultitaskprotDB: Todas las anotaciones GO de cada proteína

Puede verse el tipo de términos GOs utilizados en las anotaciones de estas proteínas

	entry	GO	Tipo	Description	canon	moon	N_Children	N_annot_50	N_annot_90	N_annot_100
0	A0QQU5	GO:0042603	COM	capsule	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	0	3	5	12
1	A0QQU5	GO:0009986	COM	cell surface	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	0	1516	2509	3290
2	A0QQU5	GO:0005576	COM	extracellular region	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	4	11733	18585	24659
3	A0QQU5	GO:0005737	COM	cytoplasm	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	28	42245	92103	128856
4	A0QQU5	GO:0016853	FUN	isomerase activity	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	265	316	826	1322
5	A0QQU5	GO:0051082	FUN	unfolded protein binding	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	0	1283	3186	4869
6	A0QQU5	GO:0016887	FUN	ATP hydrolysis activity	"Cpn60.1"	"Chaperone: Prevents misfolding and promotes t...	0	3719	7949	11239

Nota- Ver tabla EXCEL1

- MultitaskprotDB: Términos GOs mas anotados (por número de hijos)

Compruebo que los términos GOs mas anotados en multitaskprot, para cada tipo de ontologia, son términos hoja, con N_children = 0

goweight (258r × 3c)

Tipo	N_Children	COUNT(*)
COM	0	445
COM	3	152
COM	28	81
COM	19	66
COM	1	61
COM	2	42
COM	4	39
COM	213	29
COM	5	19

.....

Tipo	N_Children	COUNT(*)
FUN	0	515
FUN	1	106
FUN	2	48
FUN	30	34
FUN	172	30
FUN	3	27
FUN	4	21
FUN	10	16
FUN	130	15

.....

Tipo	N_Children	COUNT(*)
PRO	0	683
PRO	1	207
PRO	2	157
PRO	3	73
PRO	4	58
PRO	5	58
PRO	7	56
PRO	8	42

.....

- Uniref100: Términos GOs mas anotados (por número de hijos)

Compruebo que los términos GOs mas anotados en Uniref100, para cada tipo de ontologia, son términos hoja, con N_children = 0

goweight (69r × 3c)

Tipo	N_Children	COUNT(*)
COM	0	186.144
COM	28	129.649
COM	3	90.559
COM	57	56.352
COM	19	38.357
COM	6	31.006
COM	2	28.485
COM	4	28.277
COM	1	25.362

.....

Tipo	N_Children	COUNT(*)
FUN	0	518.461
FUN	1	60.937
FUN	30	51.544
FUN	2	45.068
FUN	10	34.132
FUN	3	31.826
FUN	130	26.119
FUN	4	21.558
FUN	172	16.185

.....

Tipo	N_Children	COUNT(*)
PRO	0	309.099
PRO	1	93.700
PRO	2	68.443
PRO	7	50.768
PRO	4	35.860
PRO	3	30.774
PRO	5	24.586
PRO	11	17.973

.....

4.2. Análisis por Ontologia

4.2.1. *Ontologia "FUN": Molecular Function*

- Recupero las matrices de distancia para las distintas métricas, desde los ficheros HDF5, así como la lista de GOs

Test de Mantel: Comparación de las matrices de distancia

En este caso utilizo el coeficiente de Pearson, con 10000 permutaciones

Como resultado se obtiene:

- MantelResult.r float: Veridical correlation. Pearson product-moment correlation coefficient r. r está en el rango [-1,1], donde los valores próximos a -1 indican que hay una fuerte correlación negativa, valores próximos a +1 indican que hay una fuerte correlación positiva, y valores próximos a 0 indican que no existe correlación
- MantelResult.r float: Veridical correlation
- MantelResult.p float: Empirical p-value
- MantelResult.z float: Standard score (z-score) Número de desviaciones estándar de la media
- MantelResult.correlations array: Sample correlations
- MantelResult.mean float: Mean of sample correlations
- MantelResult.std float: Standard deviation of sample correlations

- Comparo Tanimoto y gogo y cada matriz combinada

```
- ONTOLOGIA FUN TANIMOTO - GOGO
================================================================
Veridical Correlation: 0.7186137071743317
z-score: 58.258968610558746
Empirical p-value: 0.0001

- ONTOLOGIA FUN TANIMOTO - CONCURRENCIA (50)
================================================================
Veridical Correlation: 0.9323767305100334
z-score: 65.05274889395511
Empirical p-value: 0.0001

- ONTOLOGIA FUN TANIMOTO - COMBINADA MEDIA (50)
================================================================
Veridical Correlation: 0.921569471966336
z-score: 64.66805707885123
Empirical p-value: 0.0001

- ONTOLOGIA FUN TANIMOTO - COMBINADA MAXIMO (50)
================================================================
Veridical Correlation: 0.9323767305100334
z-score: 64.54437254733743
Empirical p-value: 0.0001

- ONTOLOGIA FUN TANIMOTO - COMBINADA MEDIANA (50)
================================================================
Veridical Correlation: 0.9323767305100334
z-score: 65.2025654295018
Empirical p-value: 0.0001
```

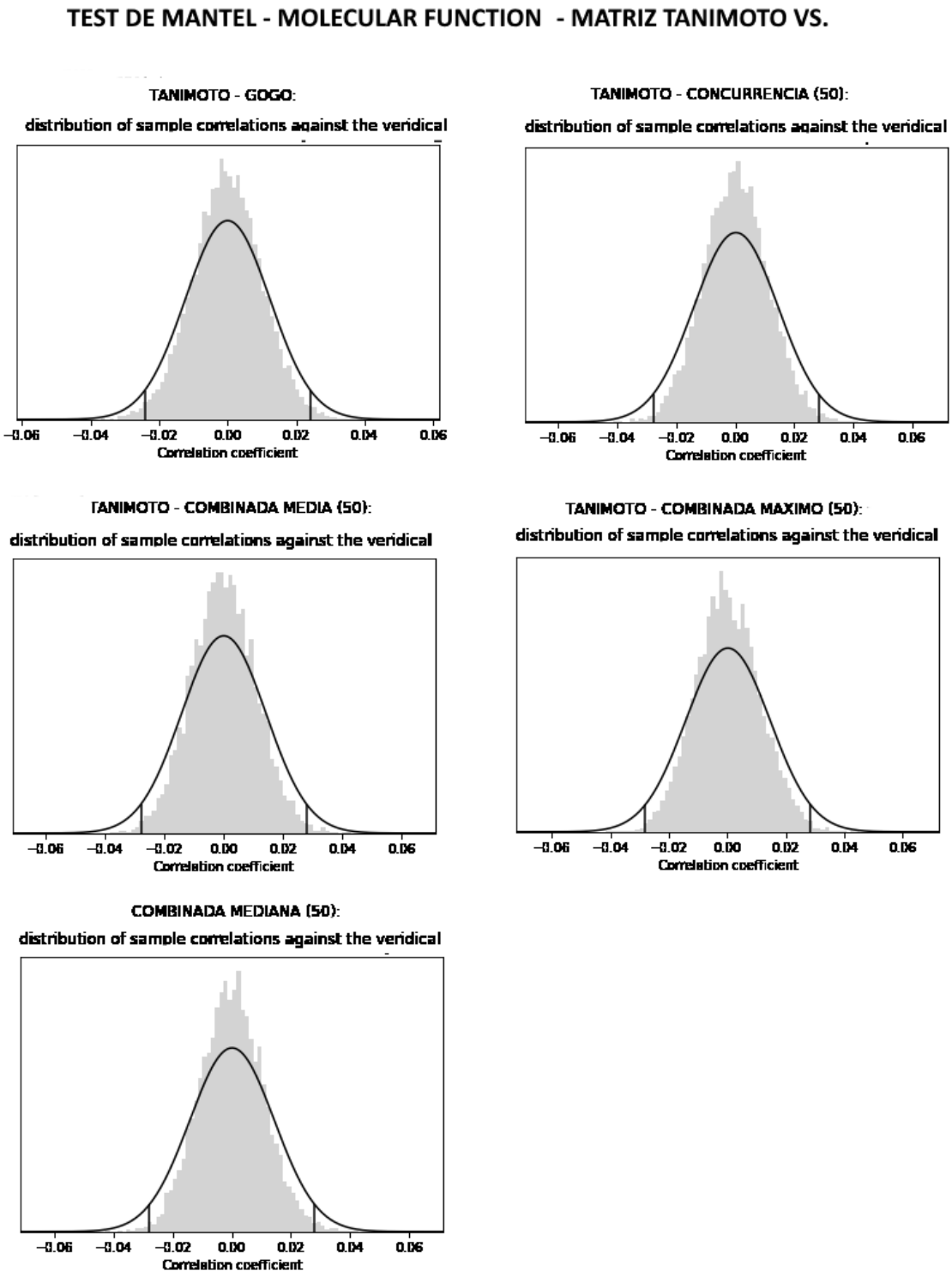


Figura 8 - FUN - Test de Mantel - Tanimoto vs

- Comparo gogo con cada matriz combinada

- ONTOLOGIA FUN GOGO - CONCURRENCIA (50)

==========

Veridical Correlation: 0.8755233511421581
z-score: 72.2004308729311
Empirical p-value: 0.0001

- ONTOLOGIA FUN GOGO - COMBINADA MEDIA (50)

==========

Veridical Correlation: 0.857202325978155

z-score: 71.49026196700625
Empirical p-value: 0.0001

- ONTOLOGIA FUN GOGO - COMBINADA MAXIMO (50)
==========
Veridical Correlation: 0.8755233511421581
z-score: 72.39466884882222
Empirical p-value: 0.0001

- ONTOLOGIA FUN GOGO - COMBINADA MEDIANA (50)
==========
Veridical Correlation: 0.8755233511421581
z-score: 72.14954858813185
Empirical p-value: 0.0001

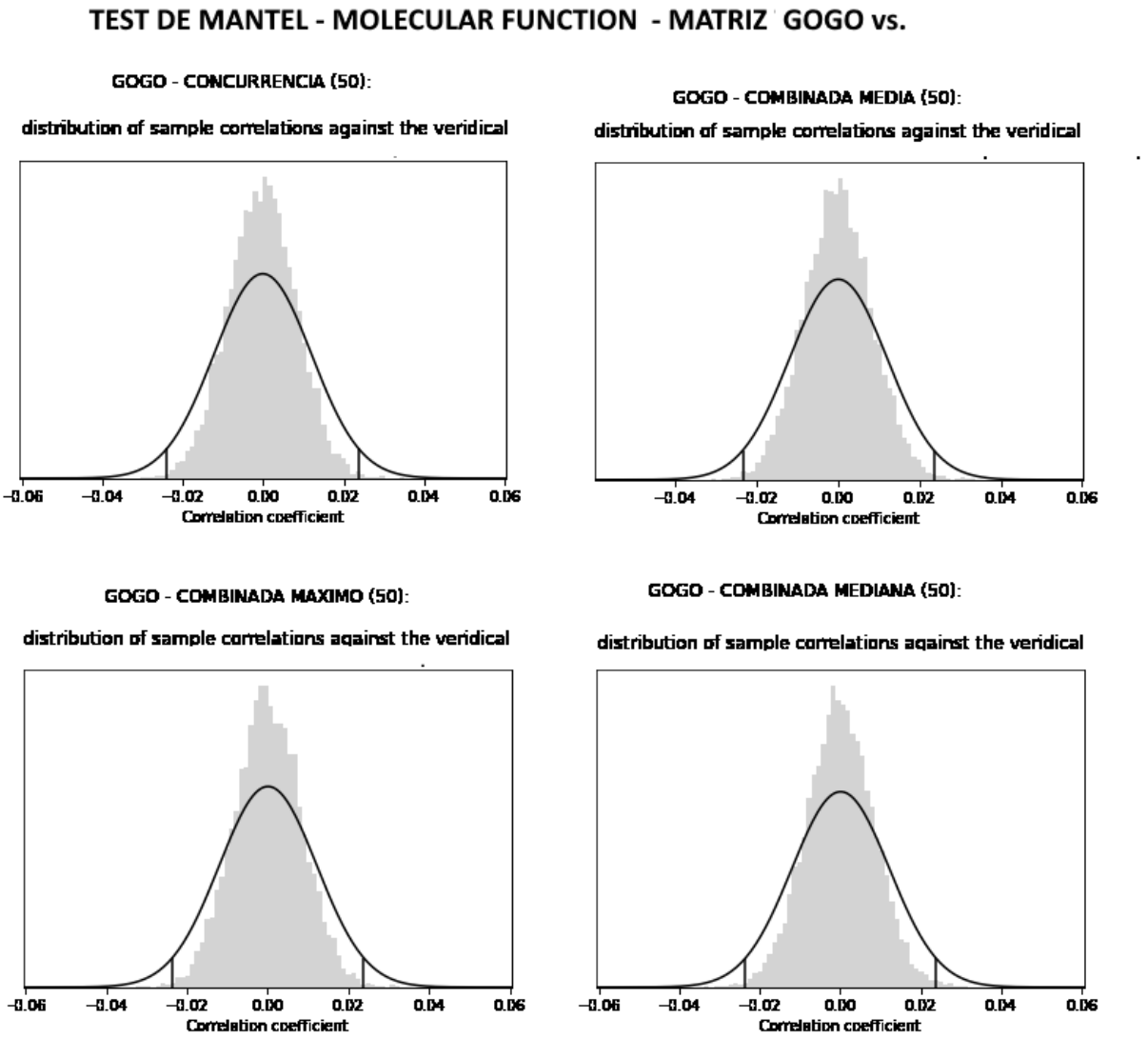


Figura 9 - FUN - Test de Mantel - GOGO vs

4.2.1.1. Heatmaps de las matrices de distancia

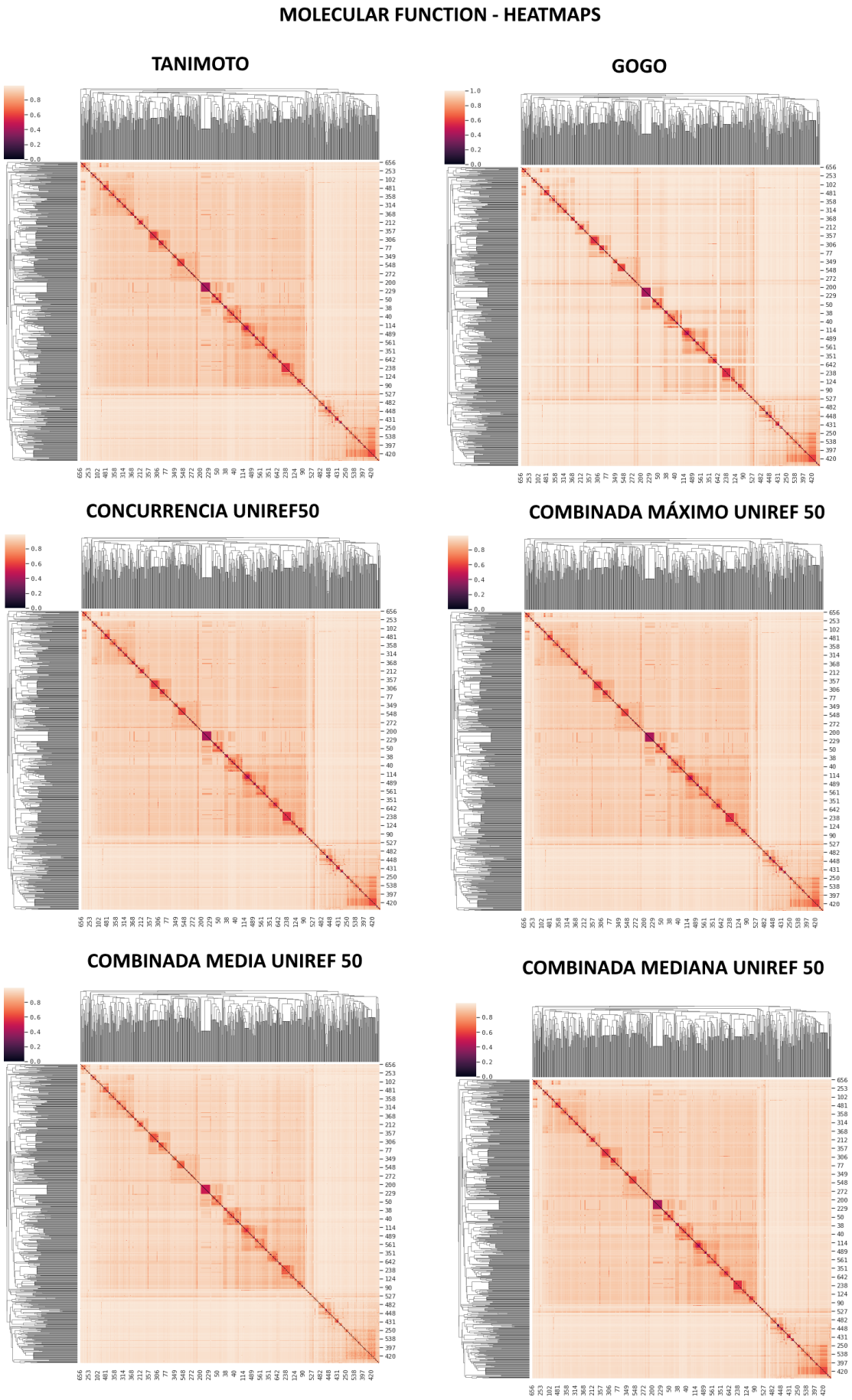


Figura 10 - FUN - Heatmaps

Las zonas más ocuras indican que los términos GO están mas cerca

4.2.1.2. Clustering

Los resultados obtenidos dependerán en gran medida de la distancia y el linkage empleados

Se generan dos modelos de clustering sobre la la matriz de distancias combinada median50., uno con linkage "complete" y otro con "average"

Hierarchical clustering: Agglometarive

```
    # Modelos
#
# ============================================================================
X_scaled = comb_median50
modelo_hclust_complete = AgglomerativeClustering(
                         affinity = 'precomputed',
                         linkage  = 'complete',
                         distance_threshold = 0,
                         n_clusters        = None
                        )
modelo_hclust_complete.fit(X=X_scaled)

modelo_hclust_average = AgglomerativeClustering(
                         affinity = 'precomputed',
                         linkage  = 'average',
                         distance_threshold = 0,
                         n_clusters        = None
                        )
modelo_hclust_average.fit(X=X_scaled)
```

Análisis del número de clusters por visualización del Dendograma

Una forma de identificar el número de clusters, es inspeccionar visualmente el dendograma de los modelos obtenidos y decidir a qué altura se corta para generar los clusters.

Visualizo el dendograma para cada modelo

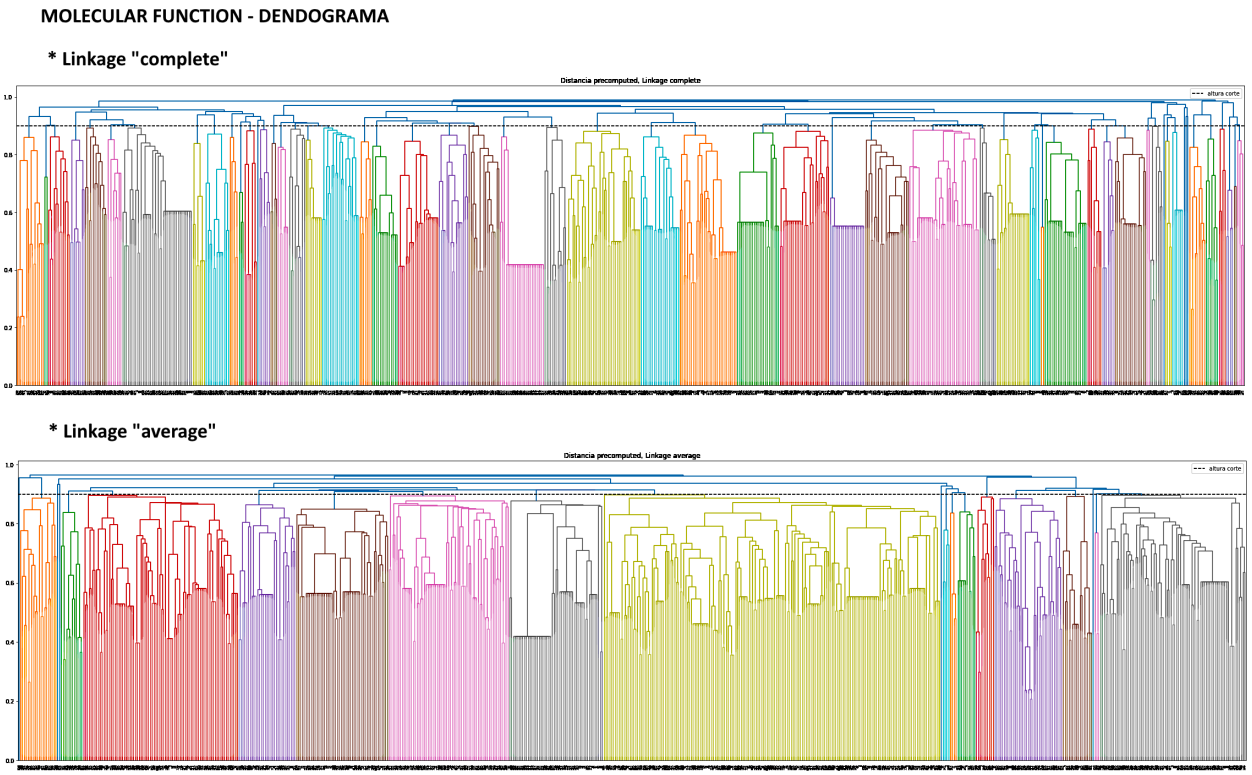


Figura 11 - FUN - Dendograma

- Linkage "complete" Obtengo alrededor de 50 clusters
- Linkage "average" Obtenemos alrededor de 18 clusters

Análisis del número de clusters : Método índices Silhouette

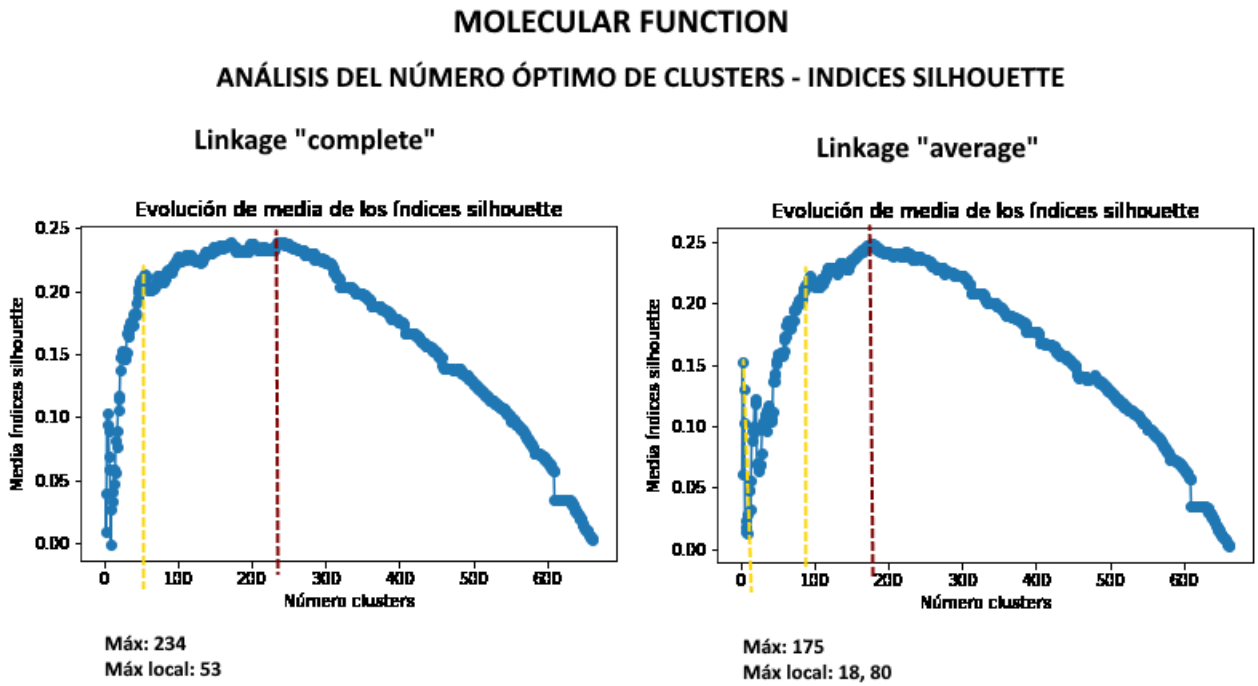


Figura 12 - FUN - Número óptimo clusters - Indices Shihouette

- Linkage "complete"

El valor máximo es :0.2394225066593583 se obtiene para el número de clusters:234

- Busco un valor menor que esté cerca del máximo, para interntar reducir el número de clusters.

Para ello observo los valores obtenidos entre 45 y 85..Para 53 hay un máximo local de 0.213, que coincide aproximadamente con los observado en el dendograma (50)

- Linkage "average"

El valor máximo es :0.2490242067082357 se obtiene para el número de clusters:175

- Busco un valor menor que esté cerca del máximo, para interntar reducir el número de clusters.

Para ello observo los valores obtenidos entre 16 y 100. En el 18 hay un máximo local, que coincide con lo observado en el dendograma, así como en el

Generación de clusters

- Genero y guardo los clusters para Linkage "complete" número de clusters 53
- Genero y guardo los clusters para Linkage "average" número de clusters 18, 53 y 92

4.2.1.3. Análisis MDS: Multidimensional Scaling

La implementación actual de MDS en scikit-learn calcula el valor del raw stress. Un valor muy alto del raw stress no indica necesariamente un mal ajuste. Por ello, calculo el stress normalizado.
[36]

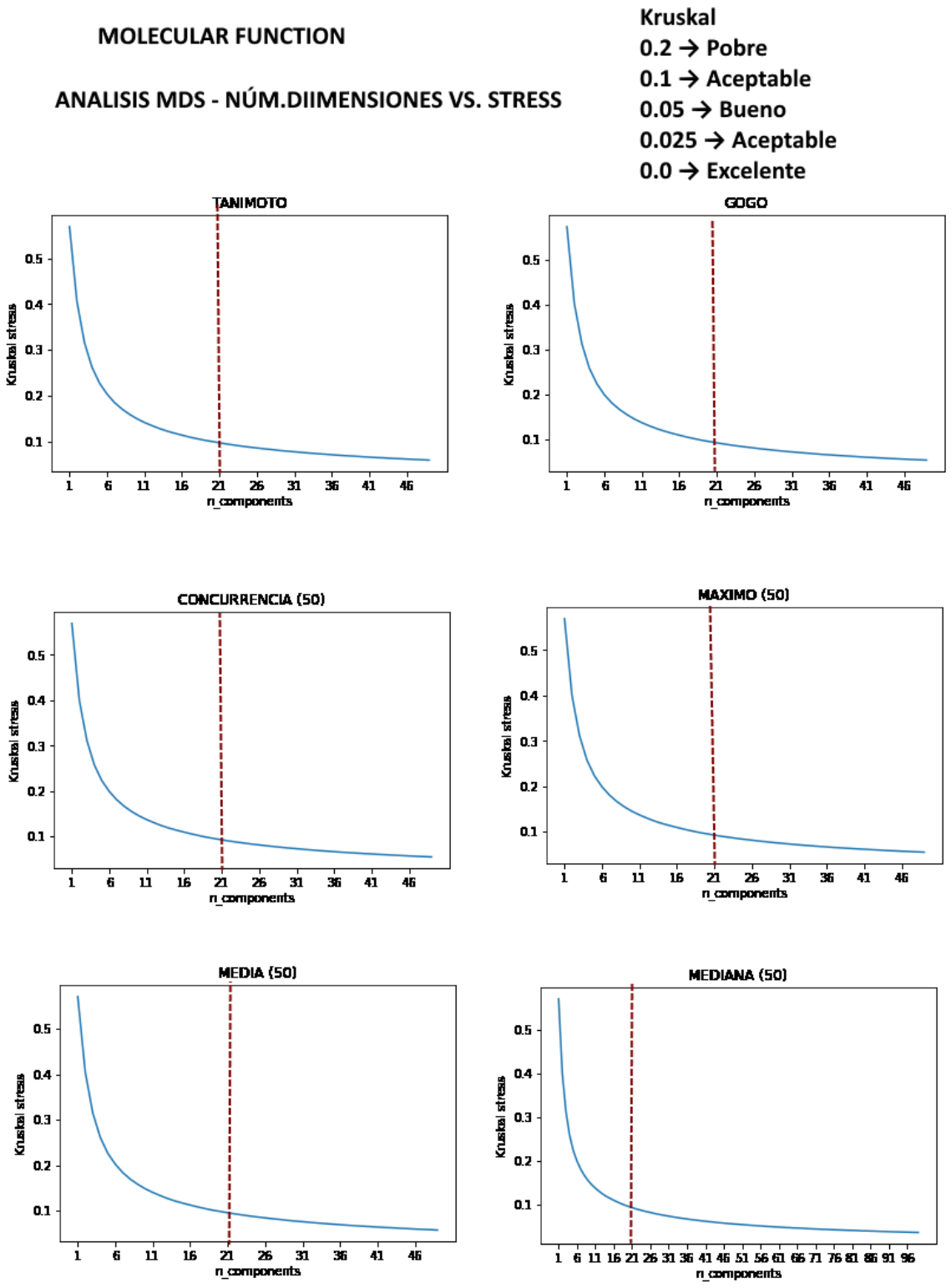


Figura 13 - FUN - Análisis MDS

- Las dimensiones óptimas para cada matriz de distancia son: (para el rango analizado, hasta 48)

Tanimoto:48
GOGO:48
CONCURRENCIA:48
MAXIMO_50:48
MEDIA_50:48

MEDIANA_50:48

Obtenemos las mismas dimensiones óptimas para las distintas matrices:

- Para n=48 con stress 0.05 (bueno)
- Para n= 21 con stress 0.1 (aceptable)

Calculo de coordenadas en el espacio reducido

- Calculo las coordenadas para la matriz de distancias median50 y para el número de dimensiones 48 (stress 0.05) y 21 (stress 0.1) y las guardo en el base de datos

4.2.1.4. Clustering K-Means sobre el espacio de coordenadas

Una vez definidos los espacios de coordenadas, hago un análisis de clusters sobre cada uno de los espacios, con el algoritmo K-Means

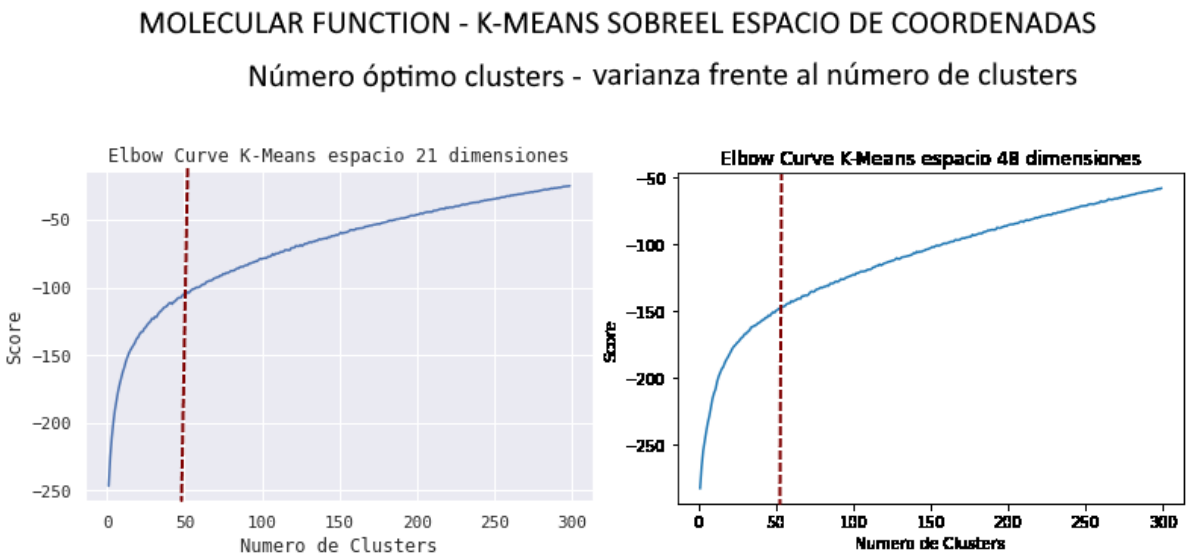


Figura 14 - FUN - K-Means

K-Means sobre el espacio de coordenadas de dimensión 21

Utilizo el método "elbow" representando la varianza frente al número de clusters
k-means score indica como de lejos están los puntos de los centroides. Los valores negativos muy grandes indican un mal score , y los valores buenos de score estarían próximos a 0
Mostrar los GOs correspondientes a estas coordenadas

Observo que para k >= 50 es cuando se produce un descenso de la varianza. Para clustering aglomerativo desde las matrices de distancia se obtenia un valor aceptable para 53 clusters

K-Means sobre el espacio de coordenadas de dimensión 48

El "codo está mas o menos en el mismo valor (>= 50), pero en este caso el score es mayor, por lo que considero que sería mejor utilizar el espacio de 21 dimensiones y número de clusters a partir de 50

Generar K-Means clusters y guardarlos en la base de datos

- Para el espacio de coordenadas de 21 dimensiones, número de clusters 53 y 92 (mismos numeros utilizados para agglomerative cluster)

NUMERO DE CLUSTERS: 53
NUMERO DE OUTLIERS: 0
NUMERO DE CLUSTERS: 92
NUMERO DE OUTLIERS: 0

4.2.1.5. Visualizar y Analizar los clusters obtenidos por diferentes métodos

- Resumen de los distintos clusters obtenidos

CLUSTERS OBTENIDOS PARA ONTOLOGIA FUN:

	Metodo	Num_Clusters
0	Agglomerative.Average	18
1	Agglomerative.Average	53
2	Agglomerative.Average	92
3	Agglomerative.Complete	53
4	K-Means.DIM21	53
5	K-Means.DIM21	92

- Visualización de los clusters obtenidos por cada método

Método Agglomerative.Average Num.Clusters 18

	CLUSTER	GO	Description
0	0	GO:0000155	phosphorelay sensor kinase activity
1	0	GO:0002935	tRNA (adenine-C2-)-methyltransferase activity
2	0	GO:0003864	3-methyl-2-oxobutanoate hydroxymethyltransfera...
3	0	GO:0003871	5-methyltetrahydropteroyltriglutamate-homocyst...
4	0	GO:0003872	6-phosphofructokinase activity
5	0	GO:0003887	DNA-directed DNA polymerase activity
6	0	GO:0003896	DNA primase activity
7	0	GO:0003951	NAD+ kinase activity
8	0	GO:0003968	RNA-directed 5-3 RNA polymerase activity
9	0	GO:0003977	UDP-N-acetylglucosamine diphosphorylase activity
10	0	GO:0003991	acetylglutamate kinase activity

.....
.....

Método K-Means.DIM21 Num.Clusters 92

	CLUSTER	GO	Description
0	0	GO:0003935	GTP cyclohydrolase II activity
1	0	GO:0004151	dihydroorotase activity
2	0	GO:0050480	imidazolonepropionase activity
3	0	GO:0102148	N-acetyl-beta-D-galactosaminidase activity
4	1	GO:0008915	lipid-A-disaccharide synthase activity
5	1	GO:0033201	alpha-1,4-glucan synthase activity
6	1	GO:0050511	undecaprenyldiphospho-muramoylpentapeptide bet...
7	1	GO:0051991	UDP-N-acetyl-D-glucosamine:N-acetylmuramoyl-L-...
8	2	GO:0042834	peptidoglycan binding
9	2	GO:1904047	S-adenosyl-L-methionine binding

.....

Nota- Ver tabla EXCEL2

4.2.1.6. Análisis Clusters sobre base de datos Multitaskprot

Utilizo la base de datos Multitaskprot para comprobar si los clusters obtenidos por los distintos métodos, son capaces de diferenciar los GOs que pertenecen a la función canónica de los GOs que pertenecen a la función Moon

- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 18
- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 53
- METODO: K-Means.DIM21, NUMERO DE CLUSTERS: 53
- METODO: K-Means.DIM21, NUMERO DE CLUSTERS: 92
-

Comparación de distintos métodos y número de clusters

- Agglomerative.Average 53 vs Agglomerative.Complete 53
- Agglomerative.Average 53 vs Agglomerative.Average 18
- Agglomerative.Average 53 vs Agglomerative.Average 92

Nota- Ver tabla EXCEL3

4.2.1.7. Análisis Clusters sobre proteínas de Uniref50

Voy a probar si los clusters obtenidos son capaces de detectar funciones diferentes sobre proteínas de Uniref50, de las que en principio desconocemos sin son o no Moonlighting
Para ello, como el conjunto de datos de todas las proteínas en muy grande, selecciono de forma aleatoria 10000 proteínas de uniref50 y muestro los clusters de los GOs anotados, para 3 tipos de clusters diferentes

- Agglomerative.Average 18
- Agglomerative.Average 53
- Agglomerative.Average 92

	entry	GO	Description	Agll_18	Aggl_53	Aggl_92
0	A0A023IWI4	GO:0090729	toxin activity	7	31	63
1	A0A023IWM6	GO:0090729	toxin activity	7	31	63
2	A0A023W168	GO:0090729	toxin activity	7	31	63
3	A0A068Q721	GO:0020037	heme binding	4	28	2
4	A0A072UR65	GO:0008061	chitin binding	1	9	25
5	A0A0A7HFE1	GO:0005524	ATP binding	4	28	2
6	A0A0G2RKY1	GO:0020037	heme binding	4	28	2
7	A0A0H2ZNL3	GO:0008760	UDP-N-acetylglucosamine 1-carboxyvinyltransfer...	3	11	33
8	A0A0H3MDW1	GO:0042803	protein homodimerization activity	10	8	22

Nota- Ver tabla EXCEL4

Observo que , generalmente, los GOs se agrupan de la misma forma en los tres tipos de cluster.

Falta comprobar, y no tengo los conocimientos para ello:

- Para las proteínas que tienen todos los GOs en los mismos clusters, si realmente esos GOs tienen la misma función.
- Para las proteínas que tienen GOs en diferentes clusters, si realmente tienen diferente función y por tanto podrían ser Moonlighting.
    - Sensibilidad: Son moonlighting y contienen GOs de diferentes clusters
    - Especificidad: No son moonlighting y todos sus GOs pertenecen al mismo cluster o a clusters co-funcion

Utilizando el mismo notebook, cambiando el parámetro «ontologia» realizo el análisis sobre las matrices de distancia de las otras dos ontologías: PRO y COM

COmo el proceso es el mismo, solo muestro los resultados obtenidos

4.2.2. Ontologia «PRO»: Biological Process

Test de Mantel : Comparación de las Matrices de Distancia

- Tanimoto vs GOGO vs Combinadas

```
- ONTOLOGIA PRO TANIMOTO - GOGO
============================================================
Veridical Correlation: 0.7663927975773221
z-score: 68.98323549287765
Empirical p-value: 0.0001

- ONTOLOGIA PRO TANIMOTO - CONCURRENCIA (50)
============================================================
Veridical Correlation: 0.8732508349706298
z-score: 75.70081756369
Empirical p-value: 0.0001

- ONTOLOGIA PRO TANIMOTO - COMBINADA MEDIA (50)
============================================================
Veridical Correlation: 0.9231420938470098
z-score: 73.67747737438137
Empirical p-value: 0.0001

- ONTOLOGIA PRO TANIMOTO - COMBINADA MAXIMO (50)
============================================================
Veridical Correlation: 0.8732508349706298
z-score: 75.82451834046212
Empirical p-value: 0.0001

- ONTOLOGIA PRO TANIMOTO - COMBINADA MEDIANA (50)
============================================================
Veridical Correlation: 0.8732508349706298
z-score: 75.4813275703272
Empirical p-value: 0.0001
```

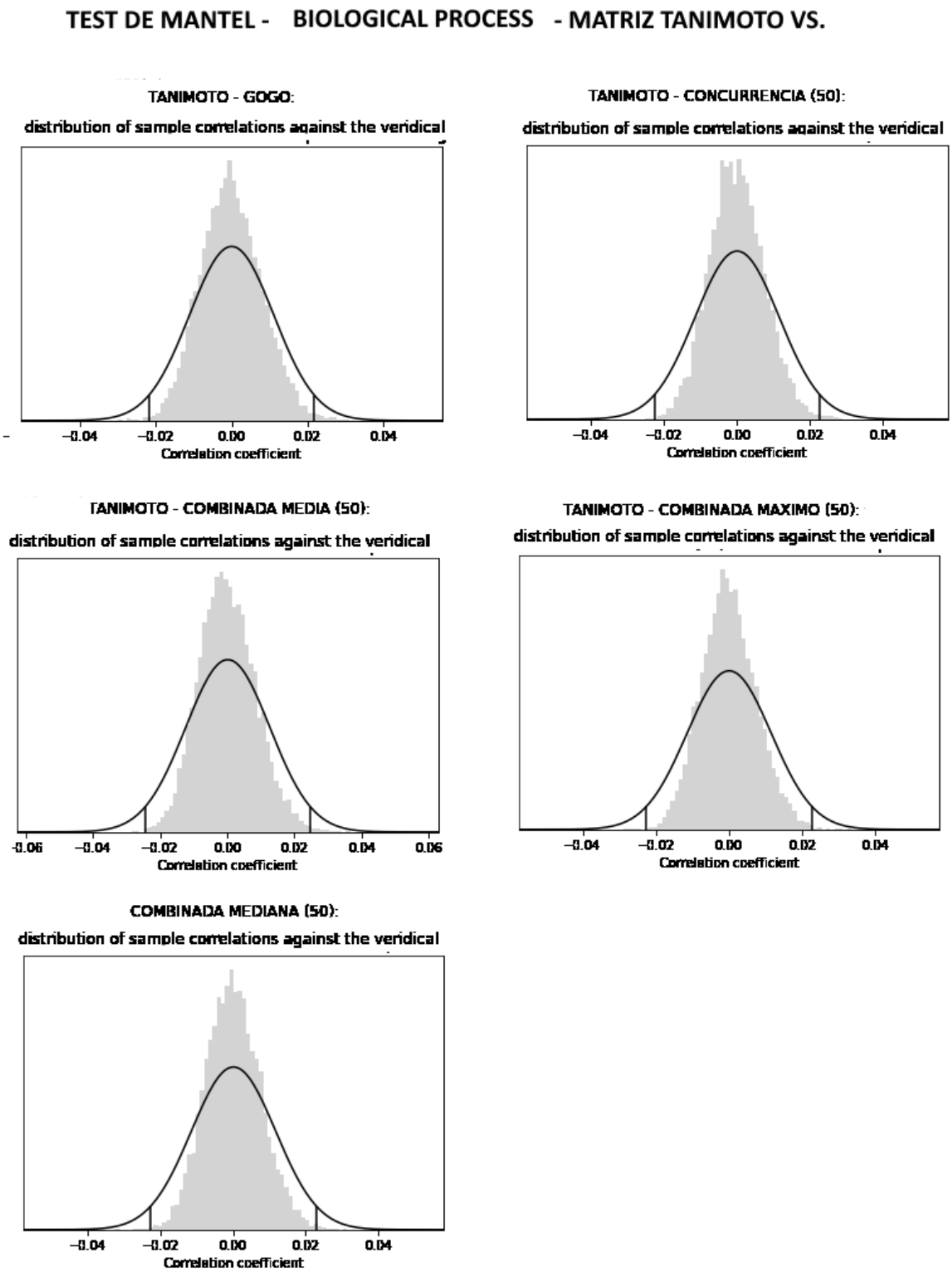


Figura 15 - PRO - Test de Mantel - Tanimoto vs

- GOGO vs combinadas

- ONTOLOGIA PRO GOGO - CONCURRENCIA (50)
==========
Veridical Correlation: 0.9579080697177416
z-score: 79.55278563842894
Empirical p-value: 0.0001

- ONTOLOGIA PRO GOGO - COMBINADA MEDIA (50)
==========
Veridical Correlation: 0.9281414508795002

z-score: 75.5055567961412
Empirical p-value: 0.0001

- ONTOLOGIA PRO GOGO - COMBINADA MAXIMO (50)
==========
Veridical Correlation: 0.9579080697177416
z-score: 79.66313276273736
Empirical p-value: 0.0001

- ONTOLOGIA PRO GOGO - COMBINADA MEDIANA (50)
==========
Veridical Correlation: 0.9579080697177416
z-score: 79.914041145447
Empirical p-value: 0.0001

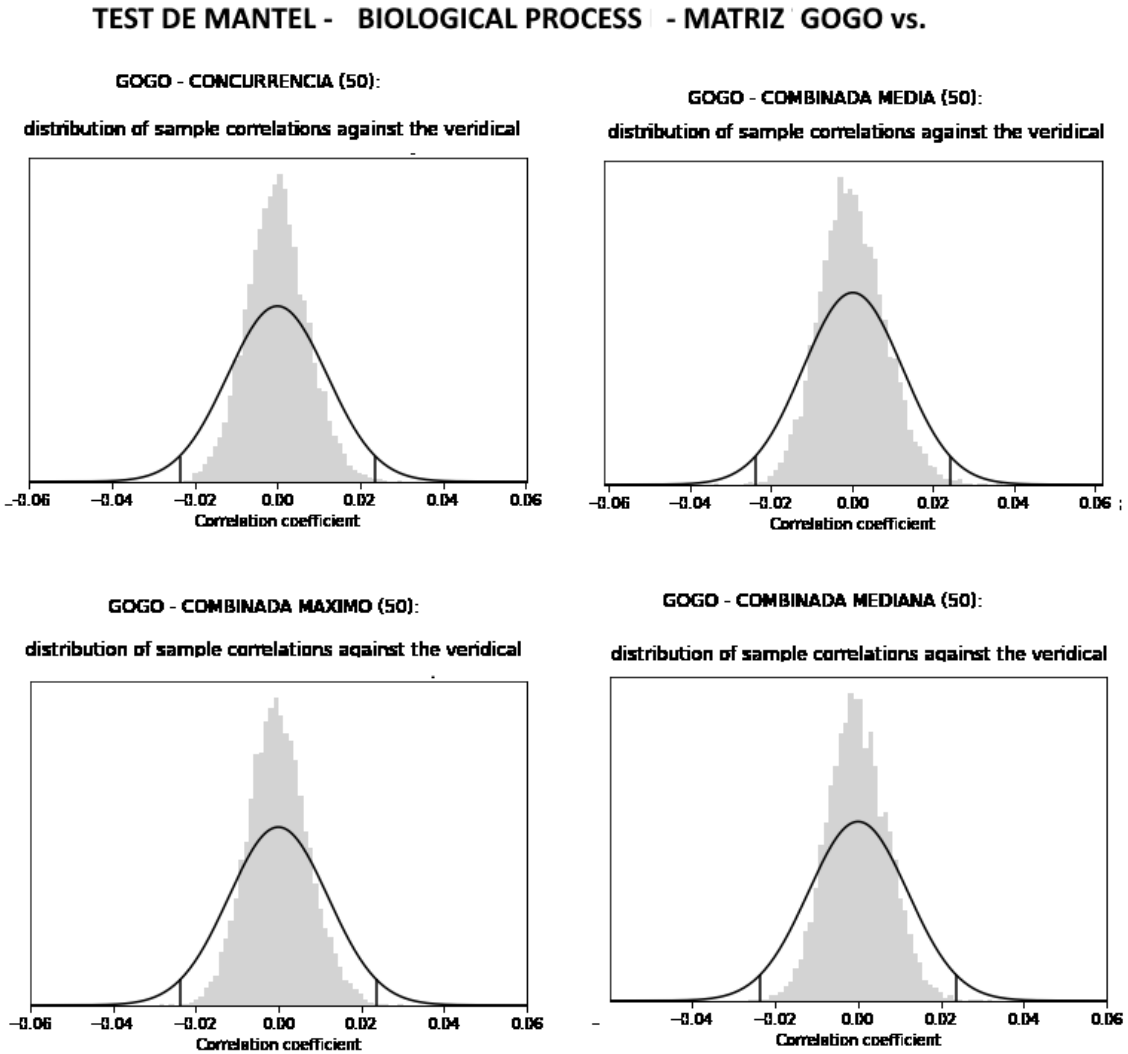


Figura 16 - PRO - Test de Mantel - GOGO vs

4.2.2.1. Heatmaps de las matrices de distancia

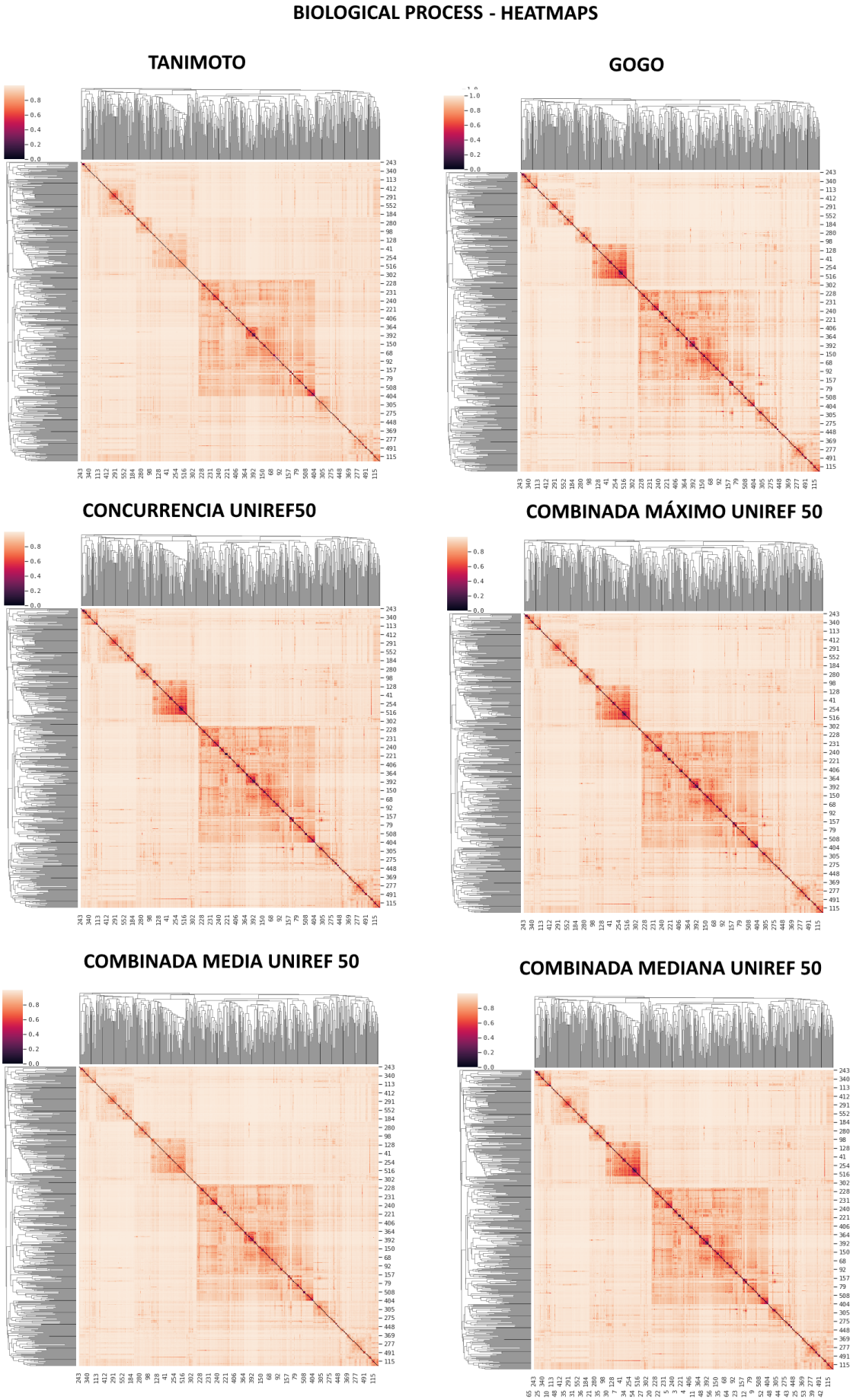


Figura 17 - PRO - Heatmaps

4.2.2.2. Clustering

Hierarchical clustering: Agglometarive

Análisis del número de clusters por visualización del Dendograma

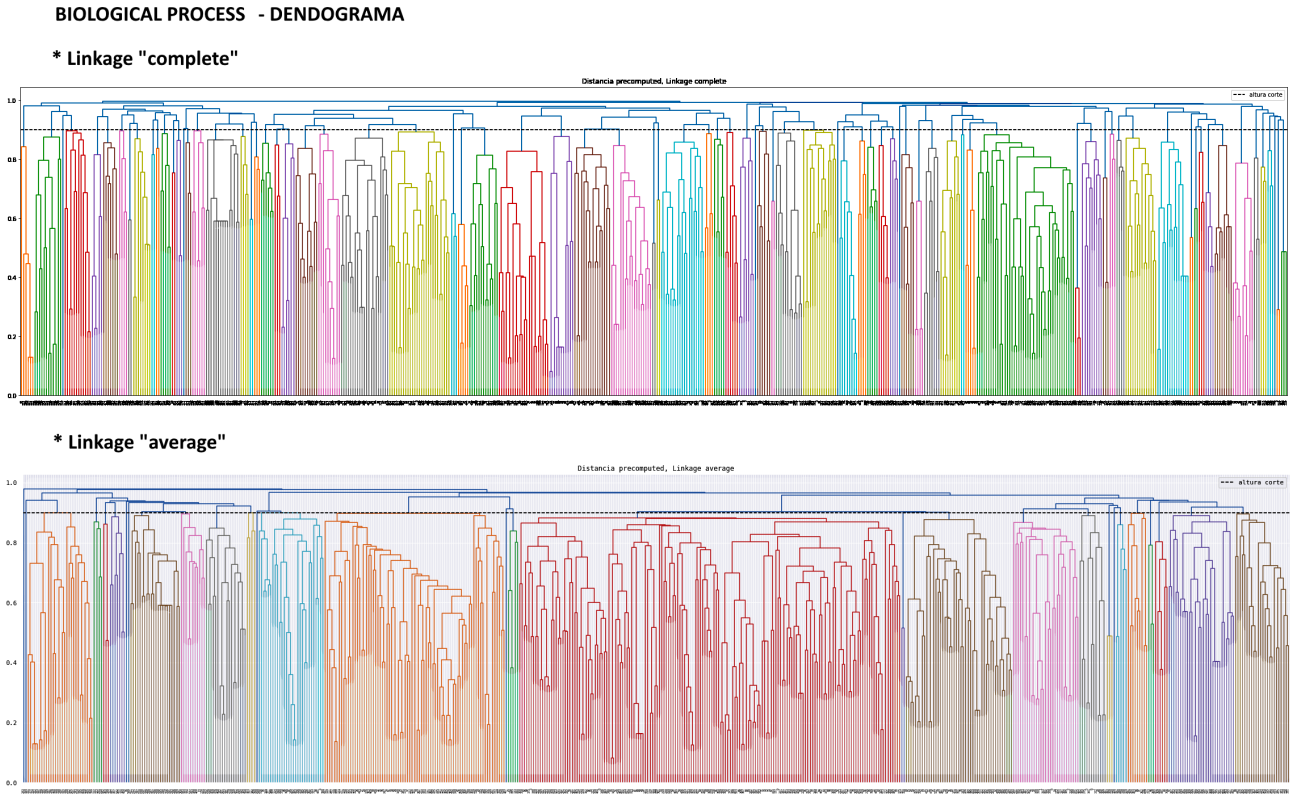


Figura 18 - PRO - Dendograma

- Linkage "complete": Obtengo alrededor de 70 clusters
- Linkage "average": Obtenemos alrededor de 23 clusters

Análisis del número de clusters : Método índices Silhouette

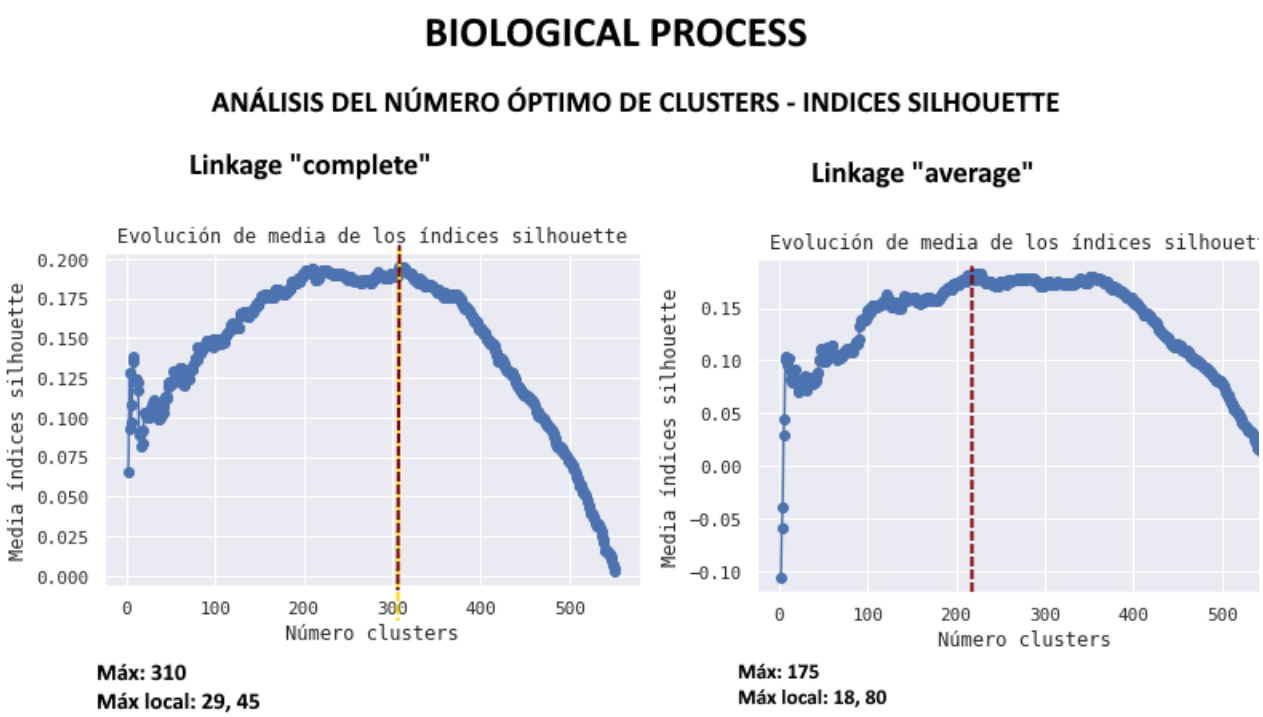


Figura 19 - PRO - Número óptimo clusters - Indices Shihouette

- Linkage "complete"

El valor máximo es :0.19478606549114458 se obtiene para el número de clusters:310

- Busco un valor menor que esté cerca del máximo, para interntar reducir el número de clusters.

Para ello observo los valores obtenidos entre 23 y 85

Obtengo máximos locales en 29 y 45, que coincide mas o menos con lo visualizado en el dendograma

- Linkage "average"

El valor máximo es :0.18270696887606994 se obtiene para el número de clusters:217

- Busco un valor menor que esté cerca del máximo, para interntar reducir el número de clusters.

Para ello observo los valores obtenidos entre 16 y 100

Hay máximos locales en 17, 21, 46, 57

Generación de clusters

- Genero y guardo los clusters para Linkage "complete" número de clusters 29 y 45
- Genero y guardo los clusters para Linkage "average" número de clusters 21,46, 57

4.2.2.3. Análisis MDS: Multidimensional Scaling

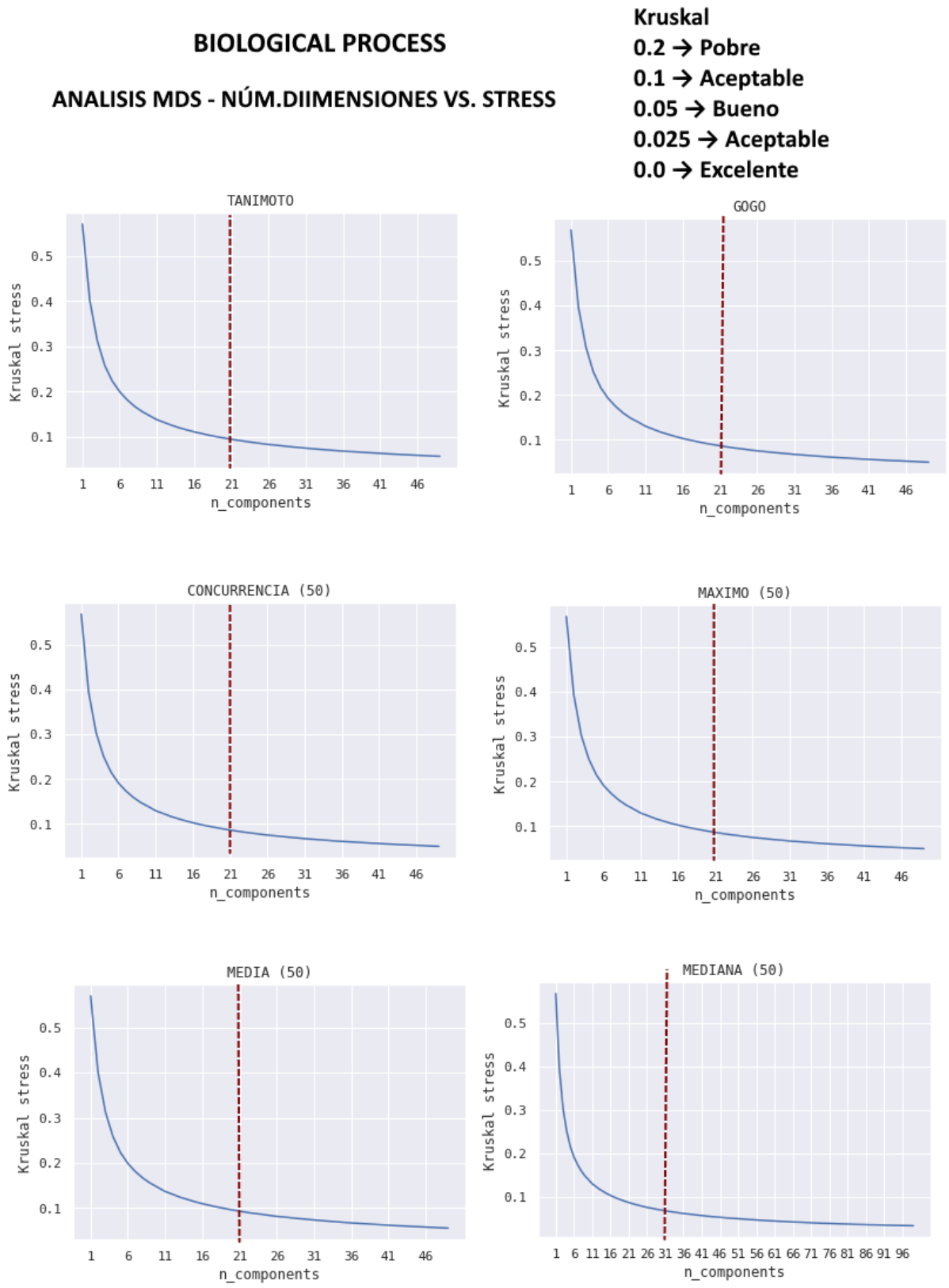


Figura 20 - PRO - Análisis MDS

- Las dimensiones óptimas para cada matriz de distancia son: (para el rango analizado, hasta 48)

Tanimoto:48
GOGO:48
CONCURRENCIA:48

MAXIMO_50:48
MEDIA_50:48
MEDIANA_50:48

Obtenemos las mismas dimensiones óptimas para las distintas matrices:

- Para n=44 con stress 0.05 (bueno)
- Para n= 21 con stress 0.1 (aceptable)

Cálculo de coordenadas en el espacio reducido

- Calculo las coordenadas para la matriz de distancias median50 y para el número de dimensiones 44 (stress 0.05) y 21 (stress 0.1) y las guardo en el base de datos

4.2.2.4. Clustering sobre el espacio de coordenadas

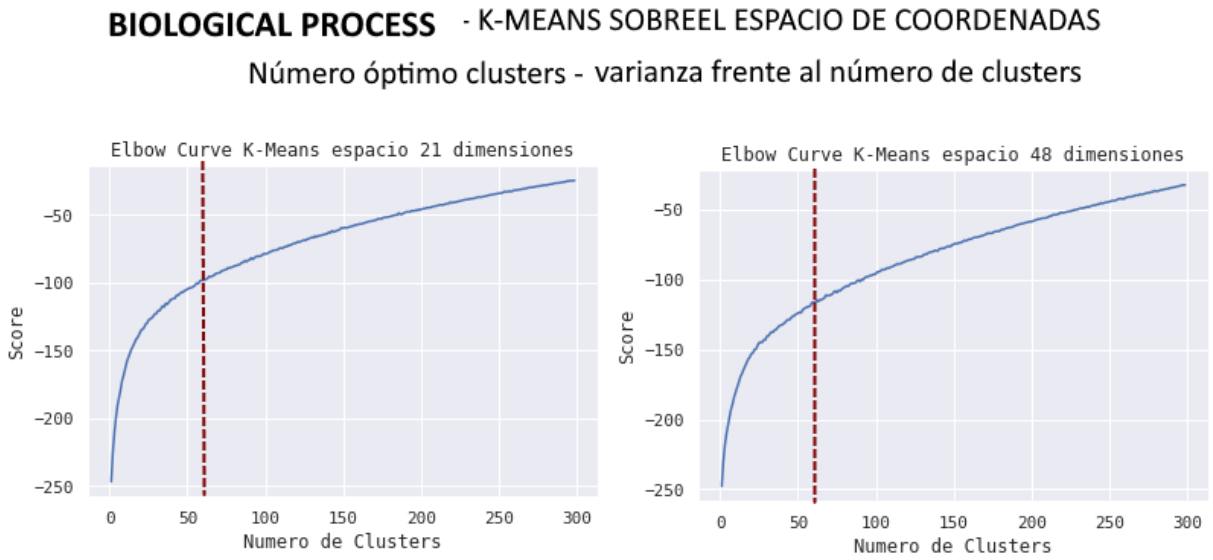


Figura 21 - PRO - K-Means

K-Means sobre el espacio de coordenadas de dimension 21

Observo que para k >= 50 es cuando se produce un descenso de la varianza. Para clustering aglomerativo desde las matrices de distancia se obtenia un valor aceptable para 57 clusters

K-Means sobre el espacio de coordenadas de dimension 44

El "codo está aproximadamente en el mismo valor (>= 50), pero en este caso el score es mayor, por lo que considero que sería mejor utilizar el espacio de 21 dimensiones y número de clusters a partir de 50, por ejemplo 57, como he utilizado para Agglomerative clustering

Generar K-Means clusters y guardarlos en la base de datos

- Para el espacio de coordenadas de 21 dimensiones, número de clusters 53 y 92 (mismos numeros utilizados para agglomerative cluster)

4.2.2.5. Visualizar y analizar los clusters obtenidos por diferentes métodos

- Resumen de los distintos clusters obtenidos

CLUSTERS OBTENIDOS PARA ONTOLOGIA PRO:

	Metodo	Num_Clusters
0	Agglomerative.Average	21
1	Agglomerative.Average	46
2	Agglomerative.Average	57
3	Agglomerative.Complete	29
4	Agglomerative.Complete	45
5	K-Means.DIM21	57

- Visualización de los clusters obtenidos por los diferentes métodos

Método Agglomerative.Average Num.Clusters 21

	CLUSTER	GO	Description
0	0	GO:0000381	regulation of alternative mRNA splicing, via s...
1	0	GO:0007190	activation of adenylate cyclase activity
2	0	GO:0007202	activation of phospholipase C activity
3	0	GO:0010575	positive regulation of vascular endothelial gr...
4	0	GO:0019228	neuronal action potential
5	0	GO:0031564	transcription antitermination
6	0	GO:0031648	protein destabilization
7	0	GO:0032689	negative regulation of interferon-gamma produc...
8	0	GO:0032691	negative regulation of interleukin-1 beta prod...
9	0	GO:0032715	negative regulation of interleukin-6 production
10	0	GO:0032720	negative regulation of tumor necrosis factor p...

.....

Nota- Ver tabla EXCEL5

4.2.2.6. Análisis Clusters sobre la base de datos Multitaskprot

- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 21
- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 46
- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 57
- METODO: Agglomerative Complete, NUMERO DE CLUSTERS: 45
- METODO: K-Means.DIM21, NUMERO DE CLUSTERS: 57

Comparación de distintos métodos y número de clusters

- Agglomerative.Average 46 vs Agglomerative.Complete 45
- Agglomerative.Average 57 vs K-Means.DIM21 57

Los clusters obtenidos, en general son equivalentes, y agrupan los mismos GOs, aunque los números de clusters obviamente difieren con cada método empleado

Nota- Ver tabla EXCEL6

4.2.2.7. Análisis Clusters sobre proteínas Uniref50

	entry	GO	Description	AgII_18	AggI_53	AggI_92
0	A0A0B4J1X5	GO:0006958	complement activation, classical pathway	2	12	27
1	A0A0K3AUJ9	GO:0070301	cellular response to hydrogen peroxide	9	5	26
2	A0A0K3AUJ9	GO:0008340	determination of adult lifespan	14	3	13
3	A0A0S4IJL0	GO:0009853	photorespiration	6	18	8
4	A0A0S4IJL0	GO:0019253	reductive pentose-phosphate cycle	6	9	12
5	A0A131MCZ8	GO:0040018	positive regulation of multicellular organism ...	0	19	4

Nota- Ver tabla EXCEL7

4.2.3. Ontología «COM»: Celular Component

Test de Mantel : Comparación de las Matrices de Distancia

- Tanimoto vs GOGO vs combinada

- ONTOLOGIA COM TANIMOTO - GOGO
==========
Veridical Correlation: 0.4322157657473386
z-score: 34.751449718469864
Empirical p-value: 0.0001

- ONTOLOGIA COM TANIMOTO - CONCURRENCIA (50)
==========
Veridical Correlation: 0.5213763427882672
z-score: 44.691171478485266
Empirical p-value: 0.0001

- ONTOLOGIA COM TANIMOTO - COMBINADA MEDIA (50)
==========
Veridical Correlation: 0.6623068052768397
z-score: 53.402164441291276
Empirical p-value: 0.0001

- ONTOLOGIA COM TANIMOTO - COMBINADA MAXIMO (50)
==========
Veridical Correlation: 0.5213763427882672
z-score: 44.74745164227595
Empirical p-value: 0.0001

- ONTOLOGIA COM TANIMOTO - COMBINADA MEDIANA (50)
==========
Veridical Correlation: 0.5213763427882672
z-score: 44.81423053741868
Empirical p-value: 0.0001

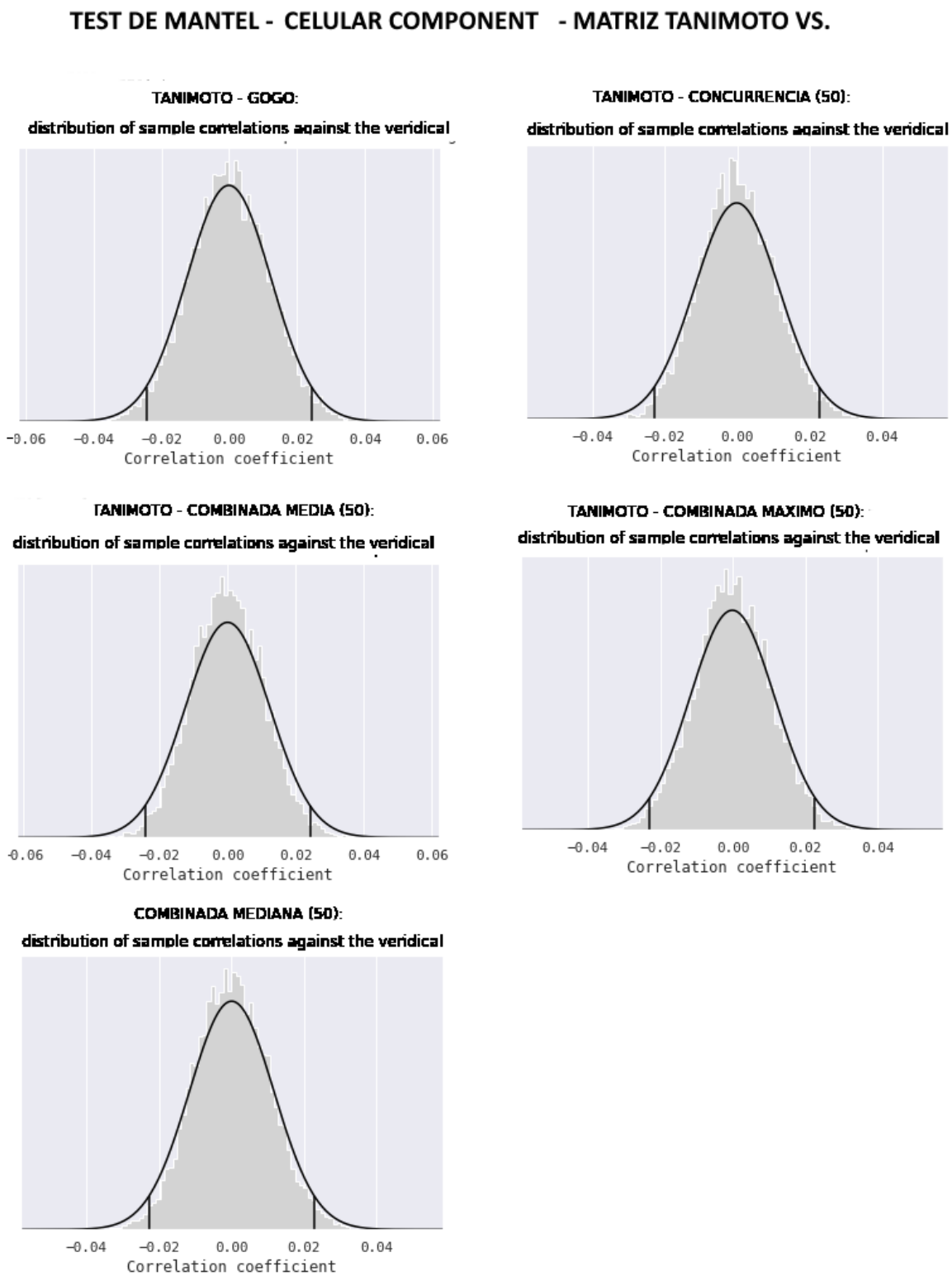


Figura 22 - COM - Test de Mantel - Tanimoto vs

- GOGO vs combinadas

- ONTOLOGIA COM GOGO - CONCURRENCIA (50)
==========
Veridical Correlation: 0.9761515440524244
z-score: 52.11636947289609
Empirical p-value: 0.0001

- ONTOLOGIA COM GOGO - COMBINADA MEDIA (50)
==========
Veridical Correlation: 0.9231095099596769

z-score: 50.20571825285496
Empirical p-value: 0.0001

- ONTOLOGIA COM GOGO - COMBINADA MAXIMO (50)
==========
Veridical Correlation: 0.9761515440524244
z-score: 52.22424375774714
Empirical p-value: 0.0001

- ONTOLOGIA COM GOGO - COMBINADA MEDIANA (50)
==========
Veridical Correlation: 0.9761515440524244
z-score: 53.02655430666994
Empirical p-value: 0.0001

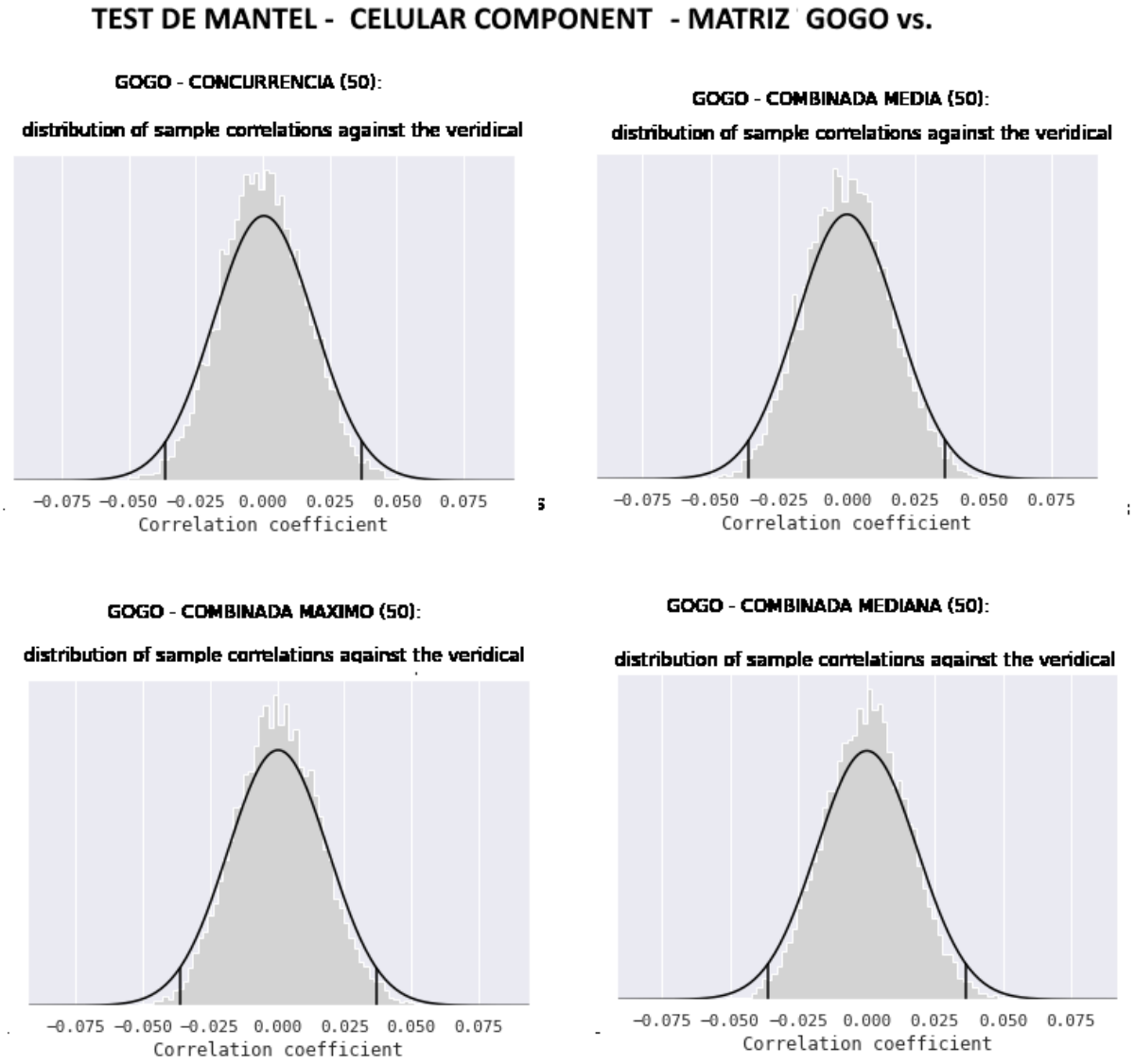


Figura 23 - COM - Test de Mantel - GOGO vs

4.2.3.1. Heatmaps de las matrices de distancia

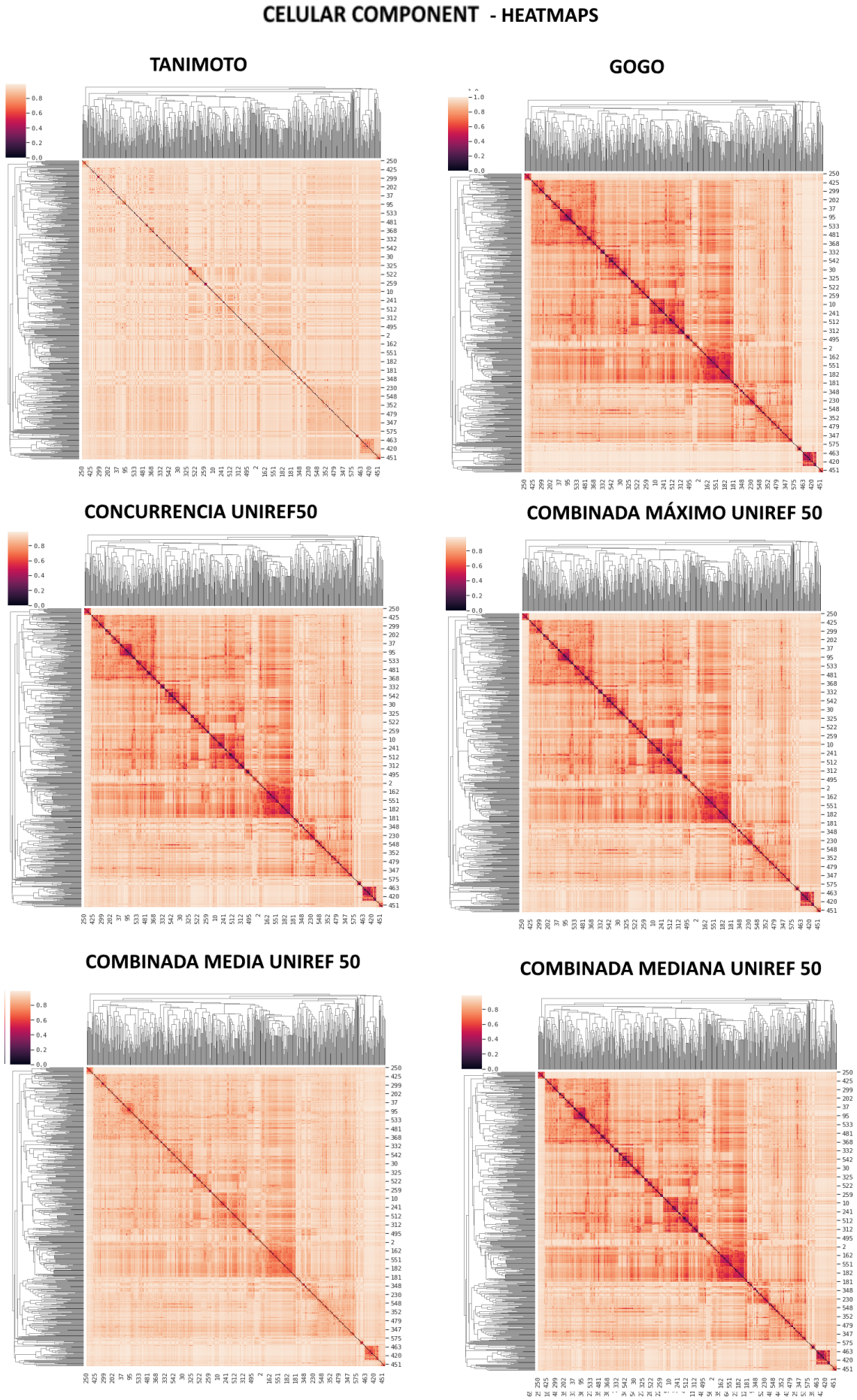


Figura 24 - COM - Heatmaps

La correlación entre la matriz Tanimoto y las matrices gogo y Concurrencia es baja, y por tanto También es baja entre Tanimoto y las matrices combinadas. Pero la correlacion enter gogo y Concurrencia es alta, así como entre gogo y Concurrencia y las matrices combinadas.
Segun los heatmaps, se detectan mas agrupaciones de GOs en gogo, Concurrencia y combinadas que en Tanimoto
Por tanto, para el clustering y MDS voy a utilizar la matriz combinada mediana Uniref 50, del mismo modo que en las otras dos ontologías

4.2.3.2. Clustering

Hierarchical clustering: Agglomerative

Para el clustering voy a utilizar la matriz de distancias combinada median50.
Genero dos modelos, uno con el método "complete" y otro con el método "average"

Análisis del número de clusters por visualización del Dendograma

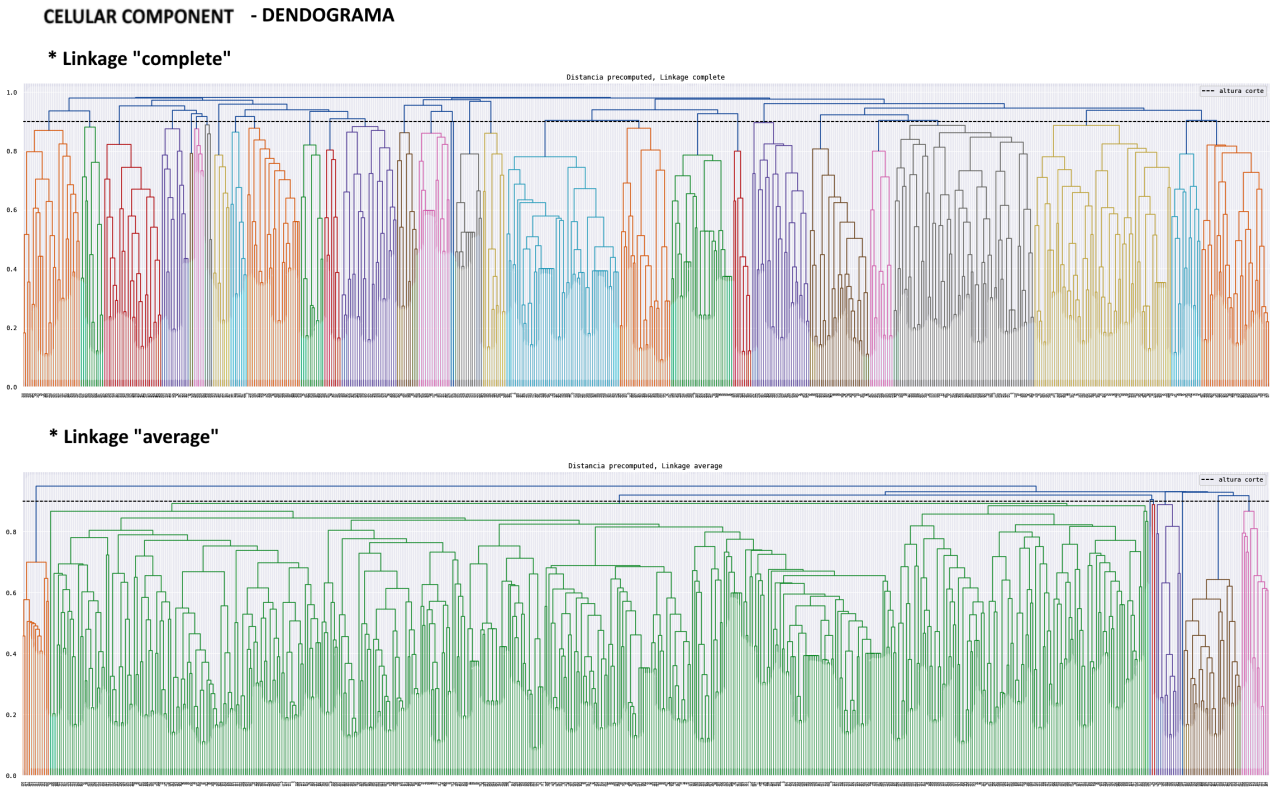


Figura 25 - COM - Dendograma

- Linkage "complete": Obtengo alrededor de 27 clusters
- Likkage «average»: Obtenemos alrededor de 6 clusters.

Análisis del número de clusters : Método índices Silhouette

- Linkage "complete"

El valor máximo es :0.17933515550812582 se obtiene para el número de clusters:158

- Busco un valor menor que esté cerca del máximo, para interntar reducir el número de clusters.

Para ello observo los valores obtenidos entre 4 y 85

Para 6 y 27 se producen máximos locales, que con los observado en el dendograma para average (6) y para complete (27)

- Linkage "average"

El valor máximo es :0.17828592026602405 se obtiene para el número de clusters:45

- Busco un valor menor que esté cerca del máximo, para intentar reducir el número de clusters.

Para ello observo los valores obtenidos entre 6 y 45
Vemos que en el 7 hay un máximo local

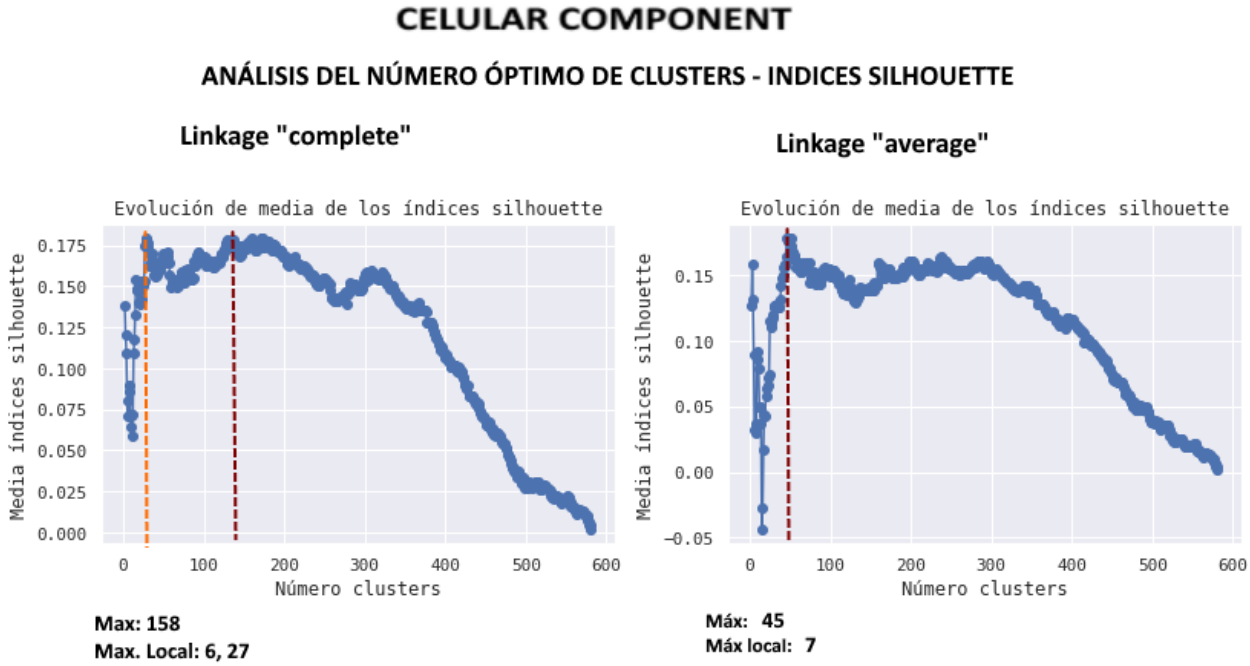


Figura 26 - COM - Número óptimo clusters - Indices Shihouette

Generación de clusters

- Genero y guardo los clusters para Linkage "complete" número de clusters 6 y 27
- Genero y guardo los clusters para Linkage "average" número de clusters 7, 45

4.2.3.3. Análisis MDS: Multidimensional Scaling

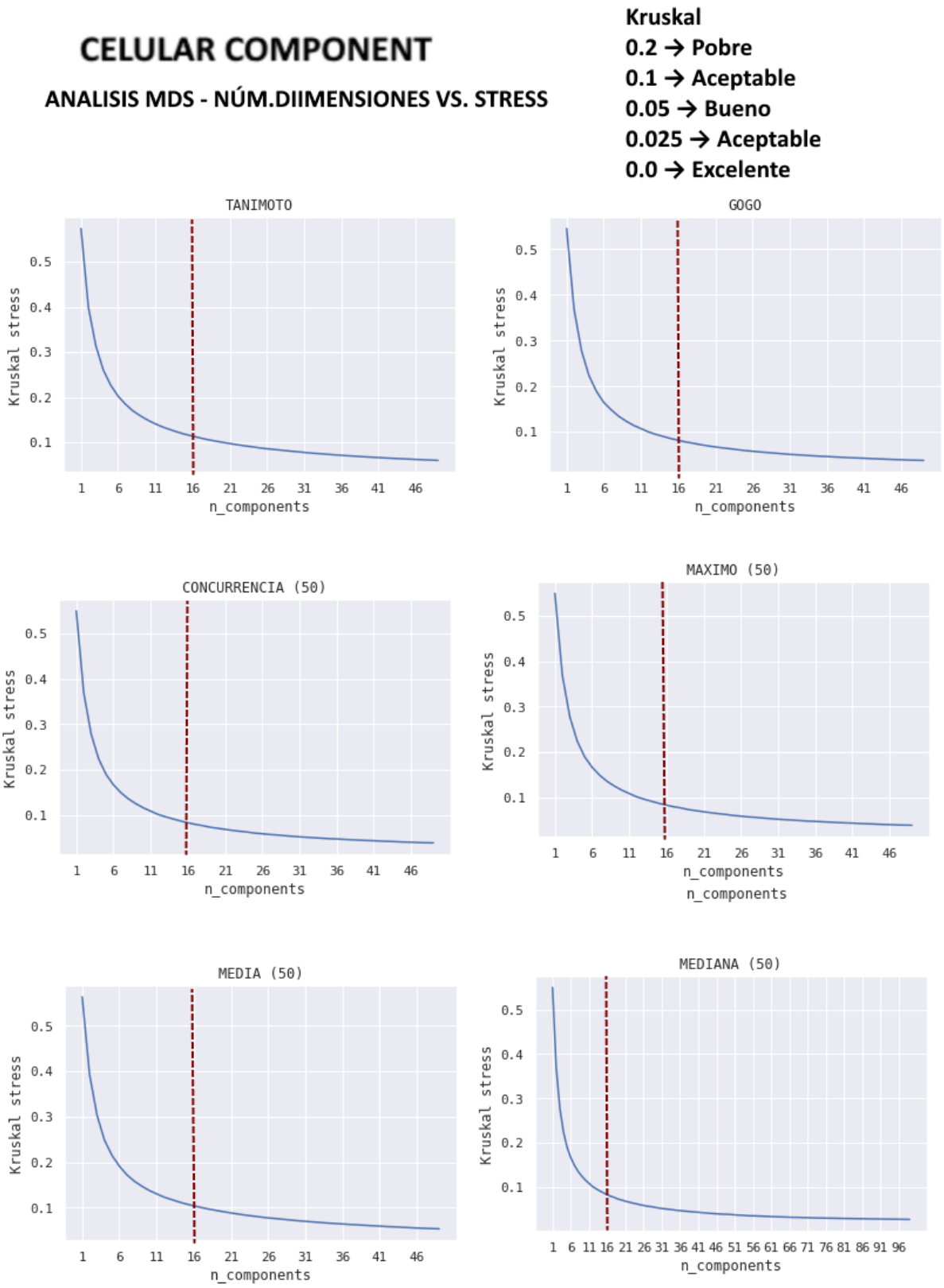


Figura 27 - COM - Análisis MDS

- Las dimensiones óptimas para cada matriz de distancia son: (para el rango analizado, hasta 48)

Tanimoto:48
GOGO:48
CONCURRENCIA:48
MAXIMO_50:48

MEDIA_50:48
MEDIANA_50:48

Obtenemos las mismas dimensiones óptimas para las distintas matrices:

- Para n=48 con stress 0.05 (bueno)
- Para n= 16 con stress 0.1 (aceptable)

Calculo de coordenadas en el espacio reducido

- Calculo las coordenadas para la matriz de distancias median50 y para el número de dimensiones 48 (stress 0.05) y 16 (stress 0.1) y las guardo en el base de datos

4.2.3.4. Clustering K-Means sobre el espacio de coordenadas

Una vez definidos los espacios de coordenadas, hago un análisis de clusters sobre cada uno de los espacios, con el algoritmo K-Means

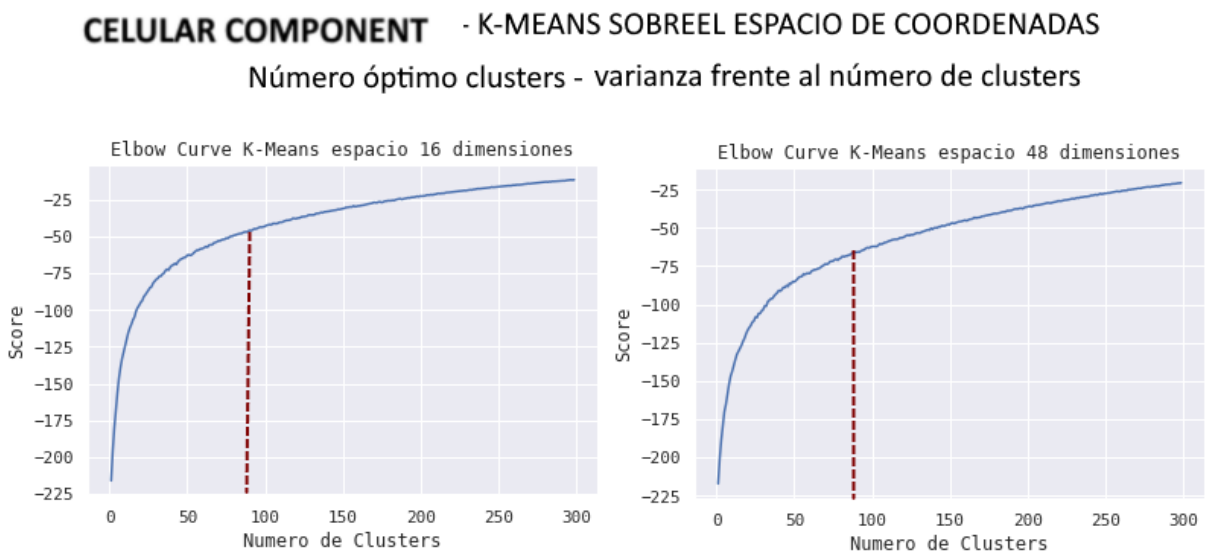


Figura 28 - COM - K_Means

K-Means sobre el espacio de coordenadas de dimension 16

Observo que para k alrededor de 70-80 es cuando se produce un descenso de la varianza. Para clustering aglomerativo desde las matrices de distancia obteniamos numeros menores, 6,27, 45

K-Means sobre el espacio de coordenadas de dimension 48

El "codo" está mas o menos en el mismo valor (>= 100), un valor mucho mas alto que para el espacio de 16 dimensiones

Generar K-Means clusters y guardarlos en la base de datos

- Para el espacio de coordenadas de 16 dimensiones, número de clusters 80

4.2.3.5. Visualizar y analizar los clusters obtenidos por diferentes métodos

- Resumen de los distintos clusters obtenidos

CLUSTERS OBTENIDOS PARA ONTOLOGIA COM:

	Metodo	Num_Clusters
0	Agglomerative.Average	7
1	Agglomerative.Average	45
2	Agglomerative.Complete	6
3	Agglomerative.Complete	27
4	K-Means.DIM16	80

- Visualización de los clusters obtenidos con cada método

Método Agglomerative.Average Num.Clusters 7

	CLUSTER	GO	Description
0	0	GO:0005615	extracellular space
1	0	GO:0016914	follicle-stimulating hormone complex
2	0	GO:0019013	viral nucleocapsid
3	0	GO:0019029	helical viral capsid
4	0	GO:0019031	viral envelope
5	0	GO:0019033	viral tegument
6	0	GO:0034361	very-low-density lipoprotein particle
7	0	GO:0034362	low-density lipoprotein particle
8	0	GO:0034363	intermediate-density lipoprotein particle
9	0	GO:0034364	high-density lipoprotein particle
10	0	GO:0034366	spherical high-density lipoprotein particle

.....

Nota- Ver tabla EXCEL8

4.2.3.6. Análisis Clusters sobre base de datos Multitaskprot

- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 7
- METODO: Agglomerative Average, NUMERO DE CLUSTERS: 45
- METODO: Agglomerative Complete, NUMERO DE CLUSTERS: 6
- METODO: Agglomerative Complete, NUMERO DE CLUSTERS: 27
- METODO: K-Means.DIM16, NUMERO DE CLUSTERS: 80

Comparación de distintos métodos y número de clusters

Los clusters obtenidos, en general son equivalentes, y agrupan los mismos GOs, aunque los números de clusters obviamente difieren con cada método empleado. Así mismo en agrupaciones con menor número de clusters, se agrupan más GOs

Nota- Ver tabla EXCEL9

4.2.3.7. Análisis Clusters sobre proteínas de Uniref50

	entry	GO	Description	Aggl_7	Aggl_45	Aggl_27	KMeans_80
0	A0A075B6S6	GO:0005615	extracellular space	0	2	12	55
1	A0A087WSX0	GO:0005615	extracellular space	0	2	12	55
2	A0A0A0MS05	GO:0042101	T cell receptor complex	3	10	8	51
3	A0A0B4J245	GO:0042101	T cell receptor complex	3	10	8	51
4	A0A0C4DH38	GO:0009897	external side of plasma membrane	3	10	8	16
5	A0A0K0K1B3	GO:0042101	T cell receptor complex	3	10	8	51
6	A0A0P0WY03	GO:0009941	chloroplast envelope	2	13	0	47

5. Discusión

El proyecto depende de los datos contenidos en bases de datos muy voluminosas, dificiles de manejar de forma local y cuya información podria no tener la calidad y/o utilidad deseada.

«Incluso las proteínas con una sola función pueden ser anotadas mediante el GO usando varios términos y al ser explorados con un criterio que sólo tenga en cuenta las ramas de la clasificación jerárquica a la que el GO pertenece estén muy separadas entre ellas. Este es el caso de las cofunciones. Estás COFUNCIONES se van a caracterizar por tener una elevadísima Concurrencia con la función principal. Al revisar anotaciones del Uniprot este hecho se verá reflejado en que las probabilidades condicionales de ambas funciones será muy distintas. Es decir el 100% de las helicasas anotadas tendrán la función ATPasa, mientras que solo un pequeño porcentaje de las ATPasas habrán sido anotadas como helicasas.

Otro que problema podemos encontrarnos en las anotaciones es la utilización de términos más o menos sinónimos. Esto puede hacer qué algunas veces el mismo tipo de proteínas sean anotadas con unos términos, otras con otros términos y algunas veces con ambos términos aunque en realidad se trate de términos que están describiendo funciones muy similares.

En el cálculo de la distancia semántica, la corrección de los métodos depende en gran medida de dos factores: la calidad de los datos de anotación y la interpretación correcta de la estructura jerárquica de una ontología. Particularmente, para los métodos que dependen del contenido de información de los términos, el ruido existente en los datos de anotación puede afectar adversamente la estimación correcta del contenido de información y traer más ruido a la similitud semántica resultante. Por ejemplo, en ontología de genes, una gran proporción de anotaciones se infiere electrónicamente por similitud de secuencia de genes u otras bases de datos de anotaciones. Si tales anotaciones inferidas deben usarse o no en el cálculo del contenido de la información es todavía una pregunta abierta. Además, algunos genes se han estudiado con mucho detalle, mientras que el conocimiento de otros es aún muy limitado. Como resultado, las anotaciones disponibles están sesgadas hacia productos genéticos muy estudiados y la calidad de las anotaciones también está sesgada. Esto puede afectar negativamente a la exactitud de los resultados parciales y finales obtenidos» (Juan Cedano)

Según los resultados obtenidos en este proyecto se obtienen matrices de distancia equivalentes. Esto induce a pensar que las distintas metricas utilizadas pueden estar realmente midiendo la distancia/similitud entre GOs

La base de datos GO tiene muchos términos y no es materialmente posible trabajar con todos. Se ha seleccionado un subconjunto para cada tipod e ontologia, que en principio, parecen ser los mas utilizados y pueden aportar una información relevante.

Los clusters obtenidos por los distintos metodos aqui reflejados, parecen similares y tienden a agrupar el mismo tipo de anotaciones. Sin embargo, mientras que los métodos Agglomerative clustering y K-MEans, asignan un cluster a cada GO; el método HDBSCAN, genera muy pocos grupos y muchos ooutiers. Esto me hace pensar que hay "ruido" en los datos, matrices de distancia, y los clusters por tanto no están nitidamente identificados.

Realmente en el trabajo realizado solo puedo concluir que hay indicios de que realmente se pueden agrupar los términos GO en grupos que representan funciones biólogicas distintas, pero no puedo concluir que realmente los clusters obtenidos en este trabajo sean los adecuados, por los siguientes motivos:

Especificidad y sensibilidad: Aunque he obtenido unas tablas con las anotaciones de multitaskprot y de Uniprot, y los clusters de cada anotacion GO segun diferentes métodos, no he podido validar, por falta de conocimientos biológicos, que realmente los clusters diferentes identifican funciones biológicas diferentes ni que la anotaciones que pertenecen a un mismo cluster realmente sean "similares". Este punto me parece relevante para poder valorar los resultados obtenidos.

Además he trabajado con un subconjunto de téminos GO, que aunque parecen ser los mas utilizados y relevantes, podría ser debido a un sesgo en las anotaciones. En este punto, creo que sería necesario incluir todos los términos GOs al menos en los clusters finales (midiendo y comparando la distancia de cada

término a los centroides de cada cluster), así como incluir por interpolación todos los términos en el espacio multidimensional., que no estaba al alcance de este trabajo, por falta de tiempo y conocimientos técnicos.

6. Conclusiones

6.1 Conclusiones

Mediante el desarrollo de este proyecto

- He tenido la oportunidad de trabajar con grandes bases de datos biológicas, que hasta ahora solo conocía de forma general y teórica.
- He afianzado conocimientos acerca de métodos de clustering y he aprendido algunas técnicas como MDS que no había estudiado en el máster
- He tenido que revisar y optimizar la programación python y SQL varias veces, para poder trabajar con grandes volúmenes de datos y operaciones repetitivas.
- He aprendido lo importancia de planificar bien los objetivos de un proyecto, que el proyecto sea realista en relación a los recursos
- Me he dado cuenta de la importancia de trabajar en un equipo multidisciplinar, porque es difícil abarcar todos conocimientos y experiencia necesarios para abordar temas complejos de investigación
- Me he centrado en conseguir un trabajo mas o menos completo, del cual pudiera extraer conclusiones y sobre todo aprender. Me ha faltado probar muchas mas cosas, mas opciones, mas conocimientos técnicos, en definitiva más tiempo
- He disfrutado la oportunidad de investigar. Buscar, aprender, probar, descubrir, equivocarme, empezar de nuevo, abstraer ideas, pensar nuevas estrategias y nuevas líneas de trabajo
- Finalmente, mediante la redacción de esta memoria, he aprendido que cuando se trabaja en un proyecto ,de investigación, no es suficiente con obtener resultados y llenar páginas de código, figuras, tablas., etc. Es necesario ordenar los resultados obtenidos,, saber abstraer las ideas relevantes y lo más importante, saber transmitirlas.

6.2. Líneas de futuro

Si tuviera la oportunidad de seguir trabajando en este proyecto, las siguientes lineas de acción serían:

En primer lugar, respecto al trabajo ya realizado, aunque requiere de conocimientos biológicos, y por el momento está fuera de mi alcance:

- Validar los clusters obtenidos, estudiando la sensibilidad y especificidad de cada conjunto obtenido por diferentes métodos. Seleccionar el (los) más óptimos
- Analizar los clusters seleccionados teniendo en cuenta sus centroides en intentar asignar a cada cluster un significado biológico, así como eliminar posibles outliers que han sido asignados a cada cluster. Estudiar la cofuncionalidad de clusters.

En segundo lugar, si no se pueden validar los clusters obtenidos, o no son significativos ni representativos, me planteraía:

- Estudiar si la distancia obtenida en el MDS es realmente equivalente a la distancia de partida (matrices de distancia
- Estudiar si el subconjunto de términos GO con los que he trabajado es el mas representativo y el óptimo. Quizás sea necesario eliminar algunos términos que no aportan diferenciación biológica y sustituirlos por otros términos aunque no sean «hojas»
- Generar un espacio métrico con todos los términos GO, uno para cada tipo de ontologia (FUN, PRO, COM), interpolando los términos no incluidos en el subconjunto de partida
- Si no se obtienen buenos resultados, revisar las métricas empleadas para la construcción de las matrices de distancia, pero asignado a cada termino GO, implementación de una función gogo propia
- Considerar técnicas de supervised machine learning

Y en tercer lugar, si se hubieran obtenido resultados óptimos o aceptables en los puntos anteriores:

- implementar un programa detector de proteínas moonlighting
- implementar un programa que dada la referencia de una proteína, o gen , genere una visión gráfica de sus funciones biológicas, con los clusters a los que pertenecen sus anotaciones, el significado biológico de cada uno de ellos, y la relación entre ellos (diferenciada, co-función, etc)

6.3. Seguimiento de la planificación

Se ha seguido la planificación y metodología planteadas desde un principio. La Metodología es adecuada, pero la limitación de tiempo y conocimientos ha hecho que no pueda llegar al objetivo final, implementar un programa detector de proteínas moonlighting.

Como en cualquier proyecto software ha sido necesario replantear, reprogramar y repetir los procesos, que en ocasiones eran complejos , sobre todo en tiempo de ejecución, debido al volumen de los datos y los algoritmos estadísticos.

A pesar de todo ello, la planificación me ha ayudado a conseguir los objetivos parciales, y a implementar dos herramientas en python, dos notebooks, que permiten poder realizar la creación de la base de datos, la estructura de las tablas, la obtención de y preparación de datos, la generación de matrices de distancia y el análisis MDS y de clusters, aplicando diferentes parámetros, añadiendo o modificando métricas, variando los criterios de selección del subconjunto de términos GO.

Así pues el seguimiento de la Metodología y planificación me han servido para establecer un esquema y unas estructuras de trabajo y sentar las bases para futuras lineas de investigación.

7. Glosario

proteínas mutitfuncionales o moonlighting: pueden desempeñar mas de una función bioquímica

función canónica : Función descubierta en primer lugar

funcion moonlighting: funciones descubiertas posteriormente (despues de la canónica)

ontologia: representación formal del conocimiento en la que los conceptos se describen por su significado y las relaciones que guardan entre ellos.

grafo: conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

DAG: tipo de grafo, dirigido y acíclico. Los arcos son unidireccionales, no existen ciclos y un nodo "hijo" puede relacionarse con diferentes nodos "padres". Los términos heredan las relaciones y propiedades de sus nodos padres.

gene ontology (GO) : herramienta bioinformática cuyo objetivo es estandarizar la representación de los genes y los atributos de sus productos génicos de todas las especies

FUN: abreviatura para referirse a las anotaciones GO del tipo "función molecular" (molecular function)

PRO: abreviatura para referirse a las anotaciones GO del tipo "proceso biologico" (biological process)

COM:: abreviatura para referirse a las anotaciones GO del tipo "componente celular" (celular component)

medidas similitud: miden el grado de semejanza entre dos anotaciones

medida distancia: mide el grado de diferencia entre dos anotaciones

matrizde distancia/similitud: matriz que expresa la distancia/similitud entre una lista de objetos, por pares

clustering: Agrupación de objetos según su distancia en un espacio mñetrico

MDS: Multidimensional Scaling. Representación de n observaciones en un espacio métrico multidimencional de p < n dimensiones

sensibilidad: capacidad para detectar proteínas moonlighting. unciones diferentes deben pertenecer a diferentes clusters

especificidad: funciones similares deben pertenecer al mismo cluster

8. Bibliografía

[1] Jeffery C.J. Moonlighting proteins. *Trends Biochem.* 1999;24:8–11
Google Scholar Crossref

[2] Hernández S.Ferragut G.,Amela.I.,Perez-Pons,J.,Piñol,J.,Mozo-Villarias,A.,Cedano, J.,Querol E. Multitask-ProtDB: a database of multitasking proteins *Nucleic Acids Res.*2014;42:D517–D520.
Google Scholar Crossref PubMed

[3] Mani M.,Chen, C.Amblee, V.,Liu,H.,Mathur,T.,Zwicke,G.,Zabad, S.,Patel,B.,Thakkar, J.,Jeffery C.J. MoonProt: a database for proteins that are known to moonlight. *Nucleic Acids Res.* 2015;43:D277–D282.
Google Scholar Crossref PubMed

[4] Chapple C.E., Robisson B., Spinelli L., Guien C., Becker E., Brun C. Extreme multifunctional proteins identified from a human protein interaction network. Nat. Commun. 2015; 6:7412.
Google Scholar Crossref PubMed

[5] The UniProt Consortium UniProt: the universal protein knowledgebase. Nucleic Acids Res. 2017; 45:D158–D169.
Crossref PubMed

[6] Henderson B., Martin A. Bacterial virulence in the moonlight: Multitasking bacterial moonlighting proteins are virulence determinants in infectious disease. Infect. Immun. 2011; 79:3476–3491.
Google Scholar Crossref PubMed

[7] Gene Ontology Overview
http://geneontology.org/docs/ontology-documentation/

[8] Gene Ontology Relations.
http://geneontology.org/docs/ontology-relations/

[9] Ontologías y su importancia en Biología.
https://bioinfo2.ugr.es/biocomputacion/wp-content/uploads/2018/11/Ontologi%CC%81as.pdf

[10] https://www.uniprot.org

[11] https://ftp.uniprot.org/pub/databases/uniprot/uniref/

[12] http://www.rcim.sld.cu/revista_21/articulo_pdf/ontologiadegenes.pdf

13] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3603583/

[14] http://www.rcim.sld.cu/revista_21/articulo_pdf/ontologiadegenes.pdf

[15] http://hmong.es/wiki/Jaccard_index

[16] https://github.com/tanghaibao/goatools/blob/main/goatools/semantic.py

[17] https://github.com/tanghaibao/goatools/blob/main/notebooks/semantic_similarity_wang.ipynb

[18] Pesquita C, Faria D, Falcao AO, Lord P, Couto FM. Semantic similarity in biomedical ontologies. PLoS Comput Biol 2009; 5: e1000443.

[19] https://digital.csic.es/bitstream/10261/3267/1/functional_distances.pdf

[20] http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/AMult/tema5am.pdf

[21] https://www.ibm.com/docs/es/spss-statistics/SaaS?topic=features-multidimensional-scaling

[22] https://medium.com/@s.javi.com/escalado-multidimensional-8c5e8d004f73

[23] https://stackabuse.com/guide-to-multidimensional-scaling-in-python-with-scikit-learn/ (para python)

[24] http://cv.uoc.edu/moduls/UW03_84003_01131/web/nwin/m1/análisis_cluster.pdf

[25] https://github.com/tanghaibao/goatools

[26] https://github.com/mojaie/pygosemsim

[27] https://github.com/MEGA-GO/MegaGO

[28] http://www.rcim.sld.cu/revista_21/articulo_pdf/ontologiadegenes.pdf

[29] http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP/MATERIALESMASTER/Mat_14_master0809multi-tema7.pdf

[30] https://es.wikipedia.org/wiki/Espacio_m%C3%A9trico

[31] https://es.wikipedia.org/wiki/%C3%8Dndice_de_Jaccard

[32] https://www.nature.com/articles/s41598-018-33219-y

[33] https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html

[34] http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP/MATERIALESMASTER/Mat_14_master0809multi-tema7.pdf

[35] https://medium.com/@s.javi.com/escalado-multidimensional-8c5e8d004f73

[36] https://stackoverflow.com/questions/36428205/stress-attribute-sklearn-manifold-mds-python

Anexo

Además de este documento, se incluyen en el proyecto, los siguientes ficheros:

- ***TABLAS EXCEL***: Ante la imposibilidad de incluir tablas muy grandes en el texto, se adjuntan a este trabajo 9 tablas excel a las que se hace referencia debajo de la visualización de las primeras líneas de la tabla

    - EXCEL1.xlsx - MultitaskprotDB: Todas las anotaciones GO de cada proteína
    - EXCEL2.xlsx - FUN: Visualización de los todos clusters obtenidos por cada método
    - EXCEL3.xlsx - FUN: Análisis Clusters sobre base de datos Multitaskprot
    - EXCEL4.xlsx - FUN: Análisis Clusters sobre proteínas de Uniref50
    - EXCEL5.xlsx - PRO: Visualizar y analizar los clusters obtenidos por diferentes métodos
    - EXCEL6.xlsx - PRO Análisis Clusters sobre la base de datos Multitaskprot
    - EXCEL7.xlsx - PRO: Análisis Clusters sobre proteínas Uniref50
    - EXCEL8.xlsx - COM: Visualizar y analizar los clusters obtenidos por diferentes métodos
    - EXCEL9.xlsx - COM:Análisis Clusters sobre base de datos Multitaskprot

- ***NOTEBOOKS JUPYTER***: Incluyen el código python desarrollado para este proyecto

    - GoTerMinator_MATRICES_DISTANCIA.ipynb
    - GoTerMinatorr_ANALISIS.ipynb

Accesibles a través del siguiente link:
https://drive.google.com/drive/folders/1beYJeJJciGXdViVE4nZ0_B64Mdz9pA2E?usp=sharing