Facultad de Matemática y Computación

Distancia matricial para análisis de clúster de aminoácidos

Tesis en opción al grado de Licenciado en Ciencias Matemáticas

Autor: **Randy Suárez Rodés**
Facultad de Matemática y Computación
Universidad de La Habana.

Tutor: **Dr. Luis Armando Salomón Hernández**
Departamento de Matemática Aplicada
Facultad de Matemática y Computación
Universidad de La Habana.

La Habana
Junio, 2014.

A mi madre.

Agradecimientos

Antes que a cualquier persona, le agradezco eternamente a mi madre por formarme como la persona que soy, por no darse por vencida hasta en los momentos más duros, por enseñarme el valor de toda palabra, acción y cosa, por ayudarme a andar, mamá, nunca habrá una forma de compensar todo lo que has sacrificado por mí. A mi padre por sus consejos, por hacerme sentir que se encuentra cerca a pesar de la distancia, te extraño mi viejo.

Agradezco a mi tutor el Dr. Luis Armando Salomón Hernández por su paciencia y sabios consejos, por la guía brindada durante todo este trabajo, por todo el tiempo que me dedicó, gracias por depositar su confianza en mí.

Quiero agradecer a mis locos amigos, por la compañía en este camino no tan sencillo, por las lindas experiencias que me han regalado, por su sincera amistad.

A Germán M. Pérez Quintana por el asesoramiento brindado en materias químicas y referencias relacionadas.

A todos los profesores que de una forma u otra contribuyeron a mi formación, en especial a la profesora Valia, al profesor Valdés y al profesor Advíncula.

A mis tíos por su amor y fiel compañía, a Johnsito y Sana por ser mis hermanitos, a Lety y Albert por estar siempre presentes, al profe Humberto por enseñarme Inglés, a Eduardito, Ania y Cuca por todo su apoyo incondicional y por su cariño, finalmente a Bb´ porque a su lado siento que todo es bello.

Resumen

En este trabajo se realiza un estudio de las cadenas laterales de los aminoácidos utilizando métodos de análisis de clusters. El objetivo principal es la obtención de una métrica en $\mathbb{R}^{m\times 3}$ que permita la aplicación de técnicas de clúster en el espacio de conformación de los aminoácidos. Se muestran algunos ejemplos a partir de la métrica propuesta. Se hace un estudio de varias distancias para estructuras tridimensionales. Se proponen medidas de similitud en el espacio matricial alejadas del enfoque usual en el espacio $\mathbb{R}^p$.

Abstract

In this work we study cluster analysis for side chains in aminoacids. The main problem here is related with the correct measure needed in $\mathbb{R}^{m\times 3}$ to apply the cluster analysis methodology. We show some examples by using the proposed measure. We study several distances for 3-Dimensional structures. We propose a similarity measure in a matrix space, this approach is quite different from the usual one used in the $\mathbb{R}^p$ space.

Índice general

Introducción

En los últimos años el número de proteínas almacenadas en bases de datos (por ejemplo, en el *Protein Data Bank* (PDB)) ha aumentado considerablemente. Este incremento ha posibilitado una mayor comprensión del comportamiento conformacional de los residuos en aminoácidos. Métodos estadísticos como el de Monte Carlo, los algoritmos genéticos y análisis de clúster, por mencionar algunos, han sido aplicados a estructuras en estas bases de datos para extraer información del comportamiento conformacional de los aminoácidos, ver por ejemplo [13], [11], [17], [8], [20] y [21].

Estos estudios resultan en librerías de conformeros (*rotamers libraries*) para cadenas laterales del aminoácido. Las librerías no son más que una colección de conformaciones de las cadenas laterales de la categoría del aminoácido, por lo que juegan un papel muy importante en la identificación, representación y predicción de secuencias de proteínas.

El enfoque más usual para usar este método es tratar al aminoácido como un punto en un espacio multidimensional de ángulos diedros [7] .

En nuestro trabajo proponemos un enfoque que mantiene la estructura del aminoácido y se estudian las cadenas laterales de los aminoácidos como estructuras matriciales. Este enfoque difiere de la teoría clásica que se utiliza en el análisis de clusters cuyas técnicas están generalmente enfocadas a espacios de vectores.

Los objetivos de esta tesis están enfocados a:

- Estudiar las distancias para estructuras espaciales.
- Aplicar métodos de clúster que permitan agrupar aminoácidos a partir de su estructura matricial.
- Construir librerías de conformeros para cadenas laterales de los aminoácidos.

La tesis tiene la siguiente estructura: el Capítulo 1 versa sobre los conceptos y resultados generales sobre análisis de clúster que se necesitan en nuestra investigación y se tocan algunos elementos de química computacional. En el Capítulo 2 se hace un estudio de las distancias para estructuras espaciales y se propone una métrica para realizar el análisis de clúster en muestras de aminoácidos a partir de su estructura matricial. En el Capítulo 3 se realizan varios experimentos a tres tipos de aminoácidos utilizando la métrica propuesta, se establecen comparaciones con otras distancias usando distintos algoritmos. Finalmente se presentan las conclusiones y recomendaciones de la investigación.

Capítulo 1

Nociones Básicas

1.1. Química Computacional

La química computacional es una rama de la química que utiliza computadoras para ayudar a resolver problemas químicos. En particular, utiliza los resultados de la química teórica, incorporados en algún software para obtener las estructuras y propiedades de moléculas y cuerpos sólidos. Mientras sus resultados normalmente complementan la información obtenida en experimentos químicos, pueden, en algunos casos, predecir fenómenos químicos. La química computacional es ampliamente utilizada en el diseño de nuevos medicamentos y materiales.

El término "química teórica" se puede definir como la descripción matemática de la química. El término "química computacional" es generalmente utilizado cuando un método matemático se encuentra tan bien desarrollado que puede ser automatizado para su implementación en una computadora, ver por ejemplo [25]. Dentro de la química computacional destacan importantes áreas:

- Almacenamiento y búsqueda de datos en entidades químicas.
- Identificar correlaciones entre estructuras químicas y propiedades.
- Enfoques computacionales para ayudar a una eficiente síntesis de componentes.
- Enfoques computacionales para diseñar moléculas que interactúen de forma específica con otras moléculas (*e.g.* diseño de fármacos).

Entre las estructuras estudiadas por esta rama se encuentran las moléculas de proteínas. Nuestro trabajo se concentra en buscar o crear librerías de conformeros para su análisis

posterior, utilizar algoritmos computacionales de clúster para crear dichas librerías, pero concentrándonos específicamente en las subestructuras que conforman las proteínas: los aminoácidos, de los que hablaremos a continuación.

Aminoácidos

Los aminoácidos son sustancias químicas que poseen diversas funciones. Dentro de ellos existe un gran grupo llamado aminoácidos proteicos o α-aminoácidos (en los que concentraremos nuestro estudio) y que son los que, unidos, conforman las moléculas de proteínas. Los diferentes aminoácidos poseen diversas y particulares funciones dentro de nuestro metabolismo, la mayoría de ellas de vital importancia, ver [22] para más detalles.

Mediante uniones peptídicas los distintos aminoácidos forman moléculas de proteínas y dependiendo de la combinación y cantidad de aminoácidos que se enlazan forman los distintos tipos de proteínas, que pueden ser desde las que forman nuestros músculos hasta numerosas enzimas.

Dentro de estos aminoácidos proteicos existen dos subgrupos. Los llamados aminoácidos esenciales y los no esenciales. Estos últimos, son aquellos que nuestro organismo no puede sintetizar por sí mismo y por ende es vital incorporarlos a nuestro sistema a través de la alimentación. Por el contrario, los esenciales son aquellos que nuestro cuerpo sí puede sintetizar por sí mismo, partiendo principalmente de otros aminoácidos.

Una definición más precisa de aminoácidos en cuanto a su estructura y composición se puede enunciar de la siguiente forma:

Definición* 1.1.1** *Un aminoácido es una molécula orgánica con un grupo amino NH_2 y un grupo carboxilo $COOH$. En los α-aminoácidos, también se unen un carbono C y un hidrógeno H, estos cuatro compuestos ya mencionados constituyen lo que se conoce como* ***backbone*. Este último va unido a una cadena, conocida como cadena lateral o cadena R, la cual tiene una estructura variable, que determina la identidad y las propiedades de cada uno de los diferentes aminoácidos.*

En la Figura 1.1 se puede observar la estructura genérica de un aminoácido. Es importante notar que cada aminoácido esta representado por coordenadas Cartesianas, lo que permite interpretar dicha estructura como una matriz de posición, siendo sus primeras cuatro filas las coordenadas correspondientes al *backbone*, y las restantes las correspondientes a la cadena R.

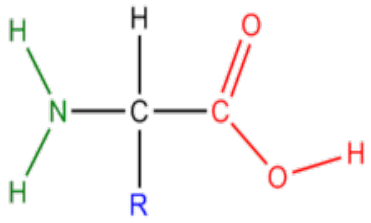


Figura 1.1: Estructura de un aminoácido

Conformación espacial y ángulos diedros

Como se había mencionado antes, entre los enfoques más usuales que se usan para estudiar la estructura de los aminoácidos en las secuencias de proteínas está aquel que se basa en la asociación de la estructura del aminoácido con un vector conformado por ángulos diedros. Para más información consultar [18] y [7].

***Definición* 1.1.2** *Un ángulo diedro no es más que el ángulo entre dos planos, es decir el ángulo que forman los respectivos vectores normales de los planos.*

Los ángulos diedros en los aminoácidos se dividen en dos categorías: ángulos relativos al *backbone* φ, ψ y ω, y los ángulos relativos a las cadenas laterales χ_i $i = 1, \cdots, 5$.

En la Tabla 1.1 se mencionan las notaciones necesarias para definir este concepto.

C	Carbono del grupo carboxilo
N	Grupo amino
H	Hidrógeno
C_α	Carbono α
C_β	Carbono β
$\vdots$	$\vdots$
R	Cadena Lateral

Tabla 1.1: Notación para los elementos de un aminoácido.

El ángulo diedro del backbone φ involucra a los átomos $C - N - C_\alpha - C$. Tomando los átomos $C - N - C_\alpha$, se determinan dos vectores, y con ellos, se calcula el vector normal al plano de estos tres átomos, de igual forma se trabaja con los átomos $N - C_\alpha - C$ y se calcula otro vector normal. Con estos dos vectores se calcula φ como se muestra en la Figura 1.2.

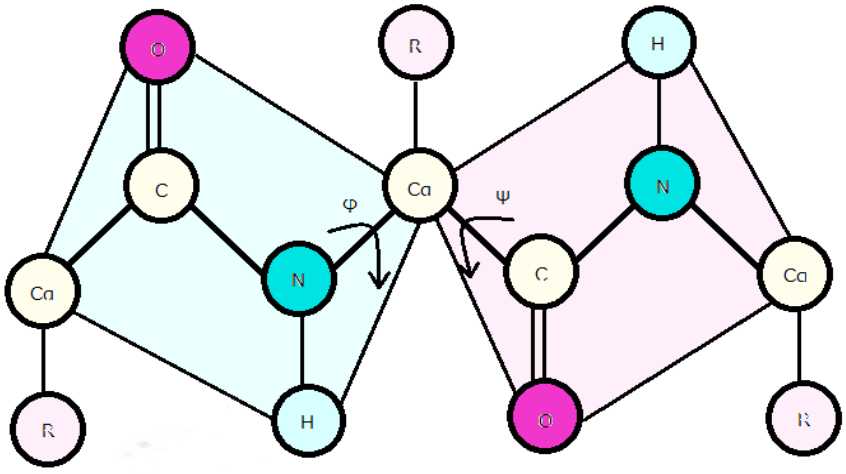


Figura 1.2: Ejemplo ilustrativo donde se observan dos ángulos diedros entre aminoácidos vecinos.

De forma análoga se procede con el ángulo ψ determinado por los átomos $N-C_\alpha-C-N$; y el ángulo ω de los átomos $C_\alpha - C - N - C_\alpha$.

Notemos que estos ángulos dependen de los vecinos del aminoácido que estamos estudiando en la secuencia de proteína que se analiza.

No dependiente del vecino pero sí del número de átomos del aminoácido están los ángulos diedros χ_i, $i = 1, \cdots, 5$ relativos a la cadena lateral.

χ_1 relativo a $N - C_\alpha - C_\beta - C_\gamma$

χ_2 relativo a $C_\alpha - C_\beta - C_\gamma - C_\delta$

χ_3 relativo a $C_\beta - C_\gamma - C_\delta - C_\epsilon$, etc.

Como se puede apreciar, el enfoque de estudio mediante ángulos diedros depende tanto de los vecinos en las secuencia de proteínas como de la cantidad de átomos en la cadena lateral. Además, este enfoque se realiza en secuencias de proteínas, usualmente no se estudia un aminoácido como estructura.

Al utilizar ángulos diedros, la estructura geométrica original de los aminoácidos, se transforma. Nuestro enfoque pretende conservar dicha estructura geométrica y realizar un análisis similar para crear una librería de conformeros.

1.2. Técnicas de Clúster

El análisis de clusters es un proceso que divide un conjunto de objetos en grupos homogéneos. Existen muchos algoritmos de clusters en publicaciones de diversas áreas como son: reconocimiento de patrones, inteligencia artificial, tecnologías de la información, procesamiento de imágenes, biología, psicología y marketing. Ver por ejemplo [10], [1], [5] y [14].

El clúster de datos constituye un componente esencial de la llamada minería de datos, un proceso que explora grandes muestras de datos con el objetivo de extraer de ella información útil.

La minería de datos se puede realizar de manera directa o indirecta. En la minería de datos indirecta ninguna variable es distinguida como un objetivo específico, y el objetivo es descubrir relaciones entre todas las variables, mientras que la minería de datos directa se fijan algunas variables como objeto de estudio. El clúster de datos es clasificado como minería de datos indirecta, ya que en él no conocemos de antemano con exactitud que clusters estamos buscando, quienes juegan un papel importante en la formación de estos clusters y como se realiza dicha formación. El problema de encontrar clusters ha sido abordado en gran medida. En general podemos adoptar la siguiente definición:

***Definición* 1.2.1** *Análisis de Clusters es un método para crear grupos de objetos de manera tal que los elementos en un mismo clúster tengan carcterísticas afines y objetos en clusters diferentes sean diferentes, según algún criterio de distancia o de similitud.*

En ese sentido es necesario obtener una matriz de similitudes o de distancias entre las variables que se desean agrupar. Cada algoritmo de clúster se basa en el índice de similitud o disimilitud entre los datos o clusters, cuando no se pueden definir entonces no es posible llevar a cabo un análisis de clusters para la muestra. Las medidas de similitud, disimilitud, o distancias son usadas para describir cuantitativamente la similitud de dos elementos y/o clusters. Estas y otras cuestiones serán abordadas con mayor profundidad a lo largo de este capítulo.

Tipos de datos

Los algoritmos de análisis de clusters están muy asociados a los tipos de datos, los cuales pueden ser clasificados como binarios, discretos o continuos.

Los datos binarios poseen exactamente dos valores, ejemplo: verdadero y falso, femenino y masculino, sí y no, etc, además pueden clasificarse en datos binarios simétricos, cuando ambos valores son igual de importantes (femenino-masculino); y por el contrario en los datos binarios asimétricos uno de los valores tiene más importancia que el otro, como "si" por la presencia de cierta propiedad y "no" por su ausencia. Datos discretos alcanzan un número finito de valores, de ahí que los datos binarios son un caso especial de dato discreto.

Las escalas de datos, las que indican la relativa significación de los datos, son también un importante punto en los algoritmos de clusters. Las mismas se dividen en escalas cualitativas, las que incluyen escalas nominales y ordinales, y las cuantitativas donde se agrupan la escala de intervalos y la escala de razón. Ver [15] para más información.

Escala de intervalos

En ellas los datos se caracterizan por ser números reales, tanto positivos como negativos, tales como, la altura, el peso, la temperatura, entre otros, los que siguen una escala lineal. Por ejemplo el tiempo entre 1905 y 1915 fue igual en duración que el tiempo entre 1967 y 1977; toma la misma cantidad de energía calentar un objeto desde $-16{,}4°C$ hasta $-12{,}4°C$ como calentarlo de $3{,}52°C$ a $39{,}2°C$. En general se requiere que el intervalo conserve el mismo nivel de importancia a lo largo de la escala.

Escala de razón

En contraste con datos en la escala de intervalos, para este tipo de datos la distinción entre 2 y 20 tiene el mismo significado que la distinción entre 20 y 200. Ejemplos típicos son la concentración de una sustancia química en cierto disolvente y la intensidad radioactiva de cierto radio isótopo. A menudo las escalas de razón se rigen por una regla exponencial en el tiempo, por ejemplo el número total de microorganismos que se multiplican en el tiempo.

Escala nominal

Estos son una generalización de los datos binarios, pero los cuales toman más de dos estados, por ejemplo en una población se pudieran considerar el color de los ojos: azules, verdes, pardos, grises, notemos que ninguno de estos atributos es más relevante que otro. En general se puede denotar el número de valores por M y etiquetamos cada uno de estos estados $1, 2, \ldots, M$, esta asignación facilita el manejo de los datos, pudiera ser 1= ojos azules, 2 = ojos pardos, 3 = ojos verdes, y 4 = ojos grises, el orden de asignación no es relevante y no caracteriza la significación de un dato sobre otro (que ojos grises tenga el 4 no significa que estos ojos sean mejores o peores que los ojos azules). Otro ejemplo sería la nacionalidad de una persona, o su estado social (soltero, casado, divorciado, viudo).

Escala ordinal

Los datos ordinales son como los nominales, solo que ahora los datos si guardan relación unos con otros, la asignación de valores $1, 2, \ldots, M$ a los datos ya no puede ser realizada de manera arbitraria, un ejemplo serían las opiniones sobre una pintura: la detesta=1, no le gusta=2, le es indiferente=3, le gusta=4, la admira=5.

Existen técnicas de conversión de una escala a otra sin perder significación de la información de los datos. Para un análisis más profundo sobre este tema consultar [4].

La naturaleza de los datos de nuestro estudio es del tipo continuo en una escala de intervalos.

Medidas de similitud y distancias

Para realizar el análisis de clusters debemos definir una medida que nos permita comparar los datos y establecer similitudes o diferencias entre los mismos con el objetivo de agruparlos, es donde necesitamos hacer uso de las medidas de similitud y disimilitud o distancias. Un coeficiente de similitud indica la fortaleza de la relación entre dos puntos. Mientras más parecidos sean dos datos, más alto será el coeficiente de similitud entre ellos.

***Definición* 1.2.2** *Sean $x = (x_1, x_2, \ldots, x_d)$, $y = (y_1, y_2, \ldots, y_d)$ dos elementos en un espacio d-dimensional, entonces el coeficiente de similitud entre x,y será una función s simétrica, $s(x, y) = s(y, x)$ que cumple:*

1. $0 \leq s(x, y) \leq 1$

2. $s(x, x) = 1$

Una medida de disimilitud o distancia se define como una métrica en el espacio en el conjunto de los datos de una muestra. Al contrario de un coeficiente de similitud, mientras más diferentes son dos datos, más alta es la distancia entre ellos. Algunos ejemplos clásicos de distancias para datos numéricos en un espacio p-dimensional se pueden apreciar en la Tabla 1.2.

Distancia	**Fórmula**
Euclideana	$d(x,y) = (\sum_i^p \lvert x_i - y_i \rvert^2)^{1/2}$
Manhattan	$d(x,y) = \sum_i^p \lvert x_i - y_i \rvert$
Máxima	$d(x,y) = \text{máx}_{1 \leq i \leq p} \lvert x_i - y_i \rvert$
Minkowski	$d(x,y) = (\sum_i^p \lvert x_i - y_i \rvert^r)^{1/r}\ r \geq 1$
Mahalanobis	$d(x,y) = \sqrt{(x-y)\Sigma^{-1}(x-y)}$
Media	$d(x,y) = (\frac{1}{p} \sum_i^p \lvert x_i - y_i \rvert^2)^{1/2}$
Chord	$2 - 2\frac{(\sum_i^p \lvert x_i - y_i \rvert^2)^{1/2}}{\lVert x \rVert \lVert y \rVert}$

Tabla 1.2: Distancias más usuales en espacios de vectores

El proceso de agrupamiento se facilita muchas veces al contar con una herramienta que nos brinde ordenadamente las relaciones entre los datos y/o clusters, una de estas herramientas es la matriz de proximidad o distancias.

Una matriz de proximidad es una matriz que contiene, relacionados dos a dos, los índices de proximidad de un conjunto de datos. Es una matriz simétrica. Por índice de proximidad entendemos tanto a un índice de similitud como de disimilitud.

Sea $D = (x_1, x_2, ..., x_n)$ un conjunto de datos, la matriz de distancia o disimilitud sería:

$$M_{dist}(D) = \begin{pmatrix} 0 & d_{12} & \cdots & d_{1n} \\ d_{12} & 0 & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & 0 \end{pmatrix}$$

De forma análoga queda determinada la matriz de similitud:

$$M_{sim}(D) = \begin{pmatrix} 1 & s_{12} & \cdots & s_{1n} \\ s_{12} & 1 & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & 1 \end{pmatrix}$$

Algoritmos

Los algoritmos se clasifican en dos categorías: *Hard Clustering* y *Fuzzy Clustering*, ver [4] para más detalles. En los primeros, un elemento pertence a un clúster y a solo uno, mientras que en *Fuzzy Clustering* las hipótesis son más suaves y un elemento puede pertenecer a uno o varios clusters con cierta probabilidad.

Nuestra investigación se concentra en técnicas de *Hard Clustering*. Los algoritmos de este enfoque se dividen en jerárquicos y particionales. Estos últimos fijan una partición para la muestra, es decir, el número de clusters a realizar es fijado desde el principio del algoritmo y no varía, mientras que los algoritmos jerárquicos dividen al conjunto de datos en una secuencia de particiones anidadas, o sea el número de clusters se va obteniendo a medida que se desarrolla el algoritmo y la creación de una nueva partición se ve ligada a las anteriores particiones.

A su vez los algoritmos jerárquicos se dividen en aglomerativos y divisivos. En los primeros el proceso comienza considerando cada dato puntual como un clúster y en cada iteración se unen el par de clusters más cercanos de acuerdo a algún criterio de similitud o disimilitud hasta que todos los datos estén en un único clúster. Por el contrario los algoritmos divisivos comienzan con todos los elementos en un clúster y continua separando elementos de los clúster superiores a clusters más pequeños.

Distancias interclusters

Como se había mencionado, muchos algoritmos son jerárquicos, estos constituyen una secuencia de particiones anidadas. Tanto en el caso de que dichos algoritmos sean aglomerativos o divisivos, es necesario calcular la distancia entre un objeto y un clúster o entre dos clusters.

En lo que sigue, sean $C_1 = \{y_1, y_2, \ldots, y_r\}$ y $C_2 = \{z_1, z_2, \ldots, z_s\}$ dos clusters, de tamaño r y s respectivamente, de una partición. También denotaremos $Dist(\cdot, \cdot)$ a la distancia entre dos clusters y $d(x, y)$ como la distancias entre dos elementos x, y del espacio muestral. Evidentemente se pueden definir muchas métricas en ese sentido. Por ejemplo

Distancia basada en la media

La disimilitud entre dos clusters en un espacio numérico, se puede determinar mediante la distancia entre las medias de dichos clusters. Sean C_1 y C_2 dos clusters en un espacio muestral numérico, la distancia media entre C_1 y C_2 viene dada por:

$$Dist_{media}(C_1, C_2) = d(\mu(C_1), \mu(C_2)),$$

donde $\mu(C_1)$, $\mu(C_2)$ son los respectivos centros de C_1 y C_2.

$$\mu(C_j) = \frac{1}{|C_j|} \sum_{x \in C_j} x.$$

Distancia del vecino más cercano, más lejano y promedio

Se pueden definir intuitivamente dos distancias entre clústers de la siguiente forma:

Dada una distancia $d(\cdot, \cdot)$ entre los datos de la muestra, la distancia del vecino más cercano (*nearest neighbor distance* (*nn*)) entre C_1 y C_2 viene dada por:

$$Dist_{nn}(C_1, C_2) = \min_{1 \leq i \leq r, 1 \leq j \leq s} d(y_i, z_j),$$

en contraste la distancia del vecino más lejano (*farthest neighbor distance* (*fn*)) se define por:

$$Dist_{fn}(C_1, C_2) = \max_{1 \leq i \leq r, 1 \leq j \leq s} d(y_i, z_j).$$

En la Figura 1.3 se aprecia un ejemplo de estas dos distancias en un espacio muestral bidimensional.

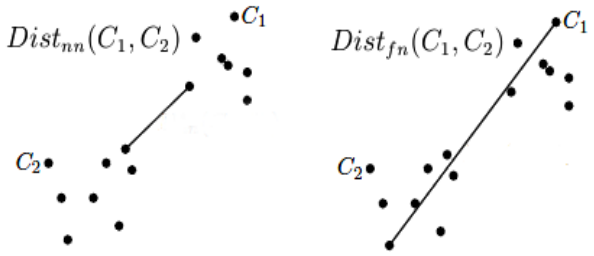


Figura 1.3: Distancia del vecino más cercano y vecino más lejano

Además se puede definir la distancia del vecino promedio(*average neighbor distance* (*ave*)) entre C_1 y C_2 respecto a $d(\cdot, \cdot)$:

$$Dist_{ave}(C_1, C_2) = \frac{1}{rs} \sum_{i=1}^{r} \sum_{j=1}^{s} d(y_i, z_j).$$

Fórmula de Lance-Williams

En los algoritmos jerárquicos se necesita calcular la distancia entre los clusters viejos y los nuevos clustres formados por dos clusters. Lance y Williams en 1967 (ver [16]) proponen una fórmula recurrente que nos permite calcular la distancia entre un cluster C_k y un cluster C formado por la fusión de dos clusters C_i y C_j, o sea $C = C_i \cup C_j$. La fórmula viene dada por:

$$\begin{aligned} Dist_{ijk}(C_k, C_i \cup C_j) &= \alpha_i Dist(C_k, C_i) + \alpha_j Dist(C_k, C_j) + \beta Dist(C_i, C_j) \\ &\quad + \gamma |Dist(C_k, C_i) - Dist(C_k, C_j)|, \end{aligned}$$

donde $Dist(\cdot, \cdot)$ representa alguna distancia intercluster definida.

Mediante una adecuada elección de los parámetros α_i, α_j, β y γ, se puede obtener varias distancias intercluster usadas en los algoritmos jerárquicos.

Métodos de agrupamiento

Para nuestro estudio emplearemos tres métodos bien conocidos del análisis de clusters, el "método del vecino más cercano", el "método de Ward", los cuales son dos algoritmos jerárquicos aglomerativos, y el método de k-means, el cual es un algoritmo particional. A continuación describiremos con un poco más de detalle cada uno de estos algoritmos.

Método Vecino más cercano

El método del Vecino más cercano o *Single Link* (SL) es uno de los algoritmos aglomerativos jerárquicos más simples. Es invariante bajo transformaciones inyectivas (por ejemplo una transformación logarítmica) sobre los datos originales. Utiliza la distancia del vecino más cercano para medir la disimilitud entre dos grupos.

Sean C_i, C_j y C_k tres grupos de datos, la distancia entre C_k y $C_i \cup C_j$ se obtiene de la fórmula de Lance-Williams haciendo $\alpha_i = \alpha_j = \frac{1}{2}$, $\beta = 0$ y $\gamma = -\frac{1}{2}$, o sea:

$$\begin{aligned} Dist(C_k, C_i \cup C_j) &= \frac{1}{2}Dist(C_k, C_i) + \frac{1}{2}Dist(C_k, C_j) - \frac{1}{2}|Dist(C_k, C_i) - Dist(C_k, C_j)| \\ &= \text{mín}\{Dist(C_k, C_i), Dist(C_k, C_j)\}, \end{aligned}$$

donde:

$$Dist(C, C') = \min_{x \in C, y \in C'} d(x, y).$$

En contraste con este método existe el método *Complete Link* el cuál utiliza la distancia del vecino más lejano, pero que no trataremos aquí.

Método de Ward

Es un procedimiento aglomerativo jerárquico que busca formar particiones $P_n, P_{n-1}, ..., P_1$ de la muestra de manera tal que se minimice la pérdida de información en cada paso de unión de los datos en los clusters.

Usualmente la pérdida de información se cuantifica en términos de un criterio de la suma de los errores al cuadrado, por lo que el método de Ward es llamado a menudo como el método de "mínima varianza".

Dado un grupo de datos C, la suma de los errores al cuadrado (*Error Sum of Squares* (ESS)) viene dada por:

$$\begin{aligned} ESS(C) &= \sum_{x \in C}(x - \mu(C))(x - \mu(C))^T \\ &= \sum_{x \in C} xx^T - \frac{1}{|C|}(\sum_{x \in C} x)(\sum_{x \in C} x)^T \\ &= \sum_{x \in C} xx^T - |C|\mu(C)\mu(C)^T, \end{aligned}$$

donde $\mu(C)$ es la media de C, es decir:

$$\mu(C) = \frac{1}{|C|} \sum_{x \in C} x.$$

Supongamos que se tienen k grupos: $C_1, C_2, ..., C_k$ entonces la pérdida de información es representada por la suma de las ESS, o sea:

$$ESS = \sum_{i}^{k} ESS(C_i).$$

En cada paso del método de Ward, la unión de cada posible par de grupos es considerada, los dos grupos cuya unión resulte en el mínimo incremento a la pérdida de información son los aglomerados en este paso.

Hasta el momento se han expuesto dos algoritmos aglomerativos clásicos de la teoría de análisis de clusters, a continuación estudiaremos el método de k-means, uno de los algoritmos particionales más populares.

Método k-means

Es uno de los algoritmos más utilizados en la teoría de clusters. Es un método particional donde se fija el número k de grupos a crear. Se procede, calculando los centros de los k clusters iniciales, se cambian los miembros de los clusters recolocando los datos a los centros más cercanos, se recalculan los centros, se vuelven a cambiar los miembros, y así sucesivamente hasta que una función de error no presente una variación significativa o los miembros de los clusters no sufran más cambios.

Sea D un conjunto de datos, y sean $C_1, C_2, ..., C_k$ los k clusters disjuntos de D. La función de error se define como:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu(C_i)),$$

donde $\mu(C_i)$ es el centro del clúster C_i.

En el esquema de la Figura 1.4 se ilustra el comportamiento del método para $k = 2$ en un espacio bidimensional.

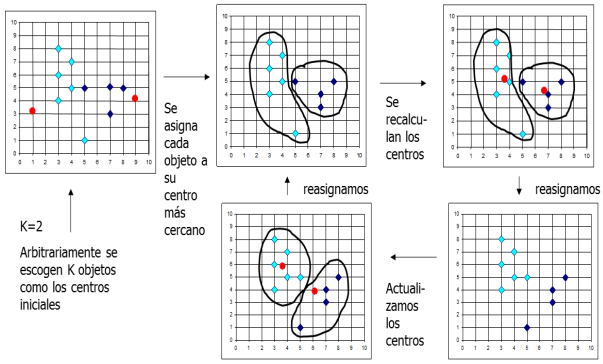


Figura 1.4: Funcionaiento del algoritmo k-means

El algoritmo de k-means posee las siguientes propiedades (ver [4]):

- Es eficiente agrupando grandes muestras de datos, debido a que su orden de complejidad computacional es lineal.
- Ofrece buenos resultados para datos numéricos.
- Su desempeño es dependiente de los centros iniciales.

Representación de algoritmos jerárquicos

Un método jerárquico es representado generalmente por un diagrama de árbol. Un n-árbol (n-*tree*) es un simple diagrama de árbol anidado que se utiliza para representar los algoritmos jerárquicos.

Sea $D = \{x_1, x_2, \ldots, x_n\}$ un conjunto. Un n-árbol sobre D se define como un conjunto $\mathcal{T}$ de subconjuntos de D que satisface las siguientes condiciones:

1 $D \in \mathcal{T}$

2 El conjunto vacío $\emptyset \in \mathcal{T}$

3 $\{x_i\} \in \mathcal{T}\ \forall i = 1, 2, \ldots, n$

4 Si $A, B \in \mathcal{T}$ entonces $A \cap B \in \{\emptyset, A, B\}$

Dendrogramas

Un dendrograma, también conocido como árbol de valores, es un n-árbol en el que a cada nodo le corresponde una altura y se cumple que:

$$h(A) \leq h(B) \Longleftrightarrow A \subseteq B$$

para todos los subconjuntos A y B con $A \cap B \neq \emptyset$

En la Figura 1.5 se muestra un dendrograma para cinco datos, claramente se puede apreciar el proceso de aglomeración de principio a fin y cada una de sus etapas. Para cada par de datos (x_i, x_j) sea h_{ij} la menor altura para la cual x_i y x_j se encuentran en un mismo grupo. Entonces un valor pequeño de h_{ij} indica una gran similitud entre (x_i, x_j). En el dendrograma brindado de ejemplo tenemos $h_{12} = 1$, $h_{14} = h_{15} = 3$ $h_{45} = 2$, y $h_{13} = 4$.

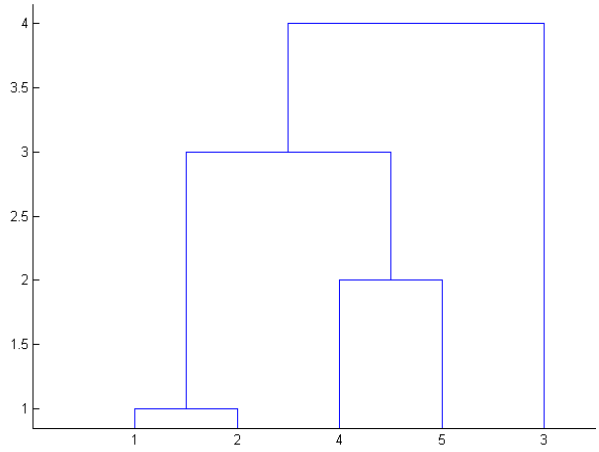


Figura 1.5: Dendrograma de 5 puntos en $\mathbb{R}^2$ con la distancia Euclideana y el método del vecino más cercano.

La altura en un dendrograma satisface la siguiente condición:

$$h_{ij} \leqslant \text{máx}(h_{ik}, h_{jk}) \ \forall \ i, j, k \in \{1, 2, \ldots, n\}.$$

De hecho, esta es una condición necesaria y suficiente para definir un dendrograma.

Usando un dendrograma podemos decidir en que momento detener el proceso de agrupamiento para obtener una cantidad deseada de clusters. En el ejemplo anterior, si cortamos el proceso antes de 3 unidades, tendríamos 2 clústers $\{x_1, x_2\}$ $\{x_4, x_5\}$ y el dato x_3 se encuentra pendiente de análisis, es decir, los datos que se encuentran a una distancia superior a 3 unidades.

Notemos que este procedimiento sólo es útil para muestras relativamente pequeñas, lo cual es una deficiencia de los algoritmos jerárquicos.

Existen otros métodos para extraer información de los algoritmos jerárquicos a partir de sus correspondientes dendrogramas como son: *banner*, representación puntual (*pointer representation*), representación en paquete (*packed representation*), entre otros que se pueden consultar en [4].

Elección del número de clusters

Existen varias estrategias para escoger un número adecuado de clusters, una de ellas como se había mencionado es haciendo uso de un dendrograma, efectuando un corte en el proceso de aglomeración, el inconveniente radica en que esta idea solo resulta viable para muestras relativamente pequeñas, existen otros criterios que no hacen uso del dendrograma como son el criterio de separación compacta (*Compactness-separation criterion*) y el criterio de la suma de cuadrados (*Sum of squares criterion*) los que deciden el número de clusters otorgando un valor a cada posible cantidad de clusters según una función de score, así el número de clusters con mayor valor es el escogido (ver [4]). Dichos métodos generalmente se aplican en datos numéricos de espacios d-dimensionales, por lo que no son conocidos estrategias análogas para datos matriciales.

Capítulo 2

Métricas para objetos en el espacio

Recordemos que la estructura de los datos a estudiar son las matrices de coordenadas de los aminoácidos. Una vez fijada dicha estructura, debemos definir una distancia para determinar el grado de similitud que poseen dos aminoácidos. Pero definir una distancia para aminoácidos nos brindará una medida de cuan alejados se encuentran espacialmente estos elementos, como se muestra en la Figura 2.1. Nuestro interés no radica realmente en hallar una medida de cercanía, si no en hallar una forma de evaluar similitudes entre estructuras, en particular, las cadenas laterales.

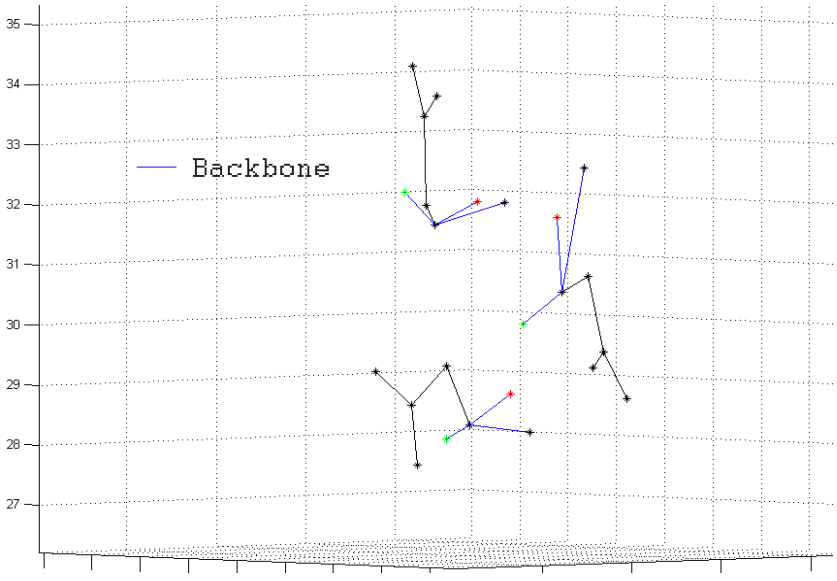


Figura 2.1: Asparagina, un α-aminoácido

2.1. Propuestas clásicas

La búsqueda de similitudes y formas en estructuras de objetos de espacios bidimensionales y tridimensionales es un problema central en ramas como visión computarizada, identificación de patrones, y en la predicción de estructuras proteicas. Muchas investigaciones se han realizado al respecto utilizando varias distancias. Distancias muy populares son la distancia de Hausdorff, la cual es muy útil para comparar conjuntos de puntos, y la distancia de Fréchet, la cual es una distancia superior para comparar cadenas poligonales. Estas dos distancias han encontrado aplicaciones muy variadas, ver por ejemplo [12], [9], [19] y [24].

Otro problema muy interesante es la superposición y el alineamiento de estructuras proteicas en un espacio tridimensional. Sobre este tema es muy utilizada la técnica de calcular la RMSD (*Root-Mean Squared Deviation*) entre dos estructuras proteicas dadas. Para más detalles ver [23].

Distancia de Hausdorff y distancia discreta de Fréchet

La distancia de Hausdorff se introdujo en 1904 por Felix Hausdorff (ver [6]), ha encontrado muchas aplicaciones como procesamiento de imágenes y reconocimiento facial, ver por ejemplo [9] y [19].

Dados dos conjuntos arbitrarios de puntos $A, B \subseteq \mathbb{R}^n$, la distancia de Hausdorff se define como:

$$d_H(A,B) = \text{máx}\{\max_{a\in A} d(a,B), \max_{b\in B} d(b,A)\},$$

donde

$$d(y,X) = \min_{x\in X} d(y,x).$$

Tomaremos la distancia de Hausdorff como una medida de disimilitud entre dos aminoácidos, considerando los mismos como el conjunto de sus puntos coordenados.

La distancia de Hausdorff es una buena medida de disimilitud entre dos conjuntos, pero es inadecuada para comparar cadenas poligonales. La distancia de Fréchet es una medida superior de disimilitud para estos casos, (ver [12] por ejemplo). Esta distancia fue introducida por Maurice Fréchet en 1906 (ver [3]). La distancia de Fréchet d_F entre dos curvas parametrizadas $f : [a,b] \longrightarrow V$ y $g : [c,d] \longrightarrow V$ donde (V,d) es un espacio métrico, viene dada, según [2], por:

$$d_F = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))),$$

donde α (respectivamente β) es una función continua no decreciente arbitraria definida en $[0,1] \longrightarrow [a,b]$ (respectivamente $[0,1] \longrightarrow [c,d]$)

La distancia de Fréchet es difícil de manejar. Eiter y Mannila (ver [2]) introdujeron una variante discreta para la distancia d_F de Fréchet.

Una curva arbitraria puede ser aproximada por una cadena poligonal. Una cadena o curva poligonal es una curva $P : [0,n] \longrightarrow V$, tal que para cada $i \in \{1,2,\ldots,n-1\}$ la restricción de P al intervalo $[i,i+1]$ es afín, o sea $P(i+\lambda) = (1-\lambda)P(i) + \lambda P(i+1)$.

Denotamos la secuencia $\{P(0), P(1), \ldots, P(n)\}$ de los puntos finales de los segmentos de P por $\sigma(P)$.

Sean P y Q dos cadenas poligonales, y $\sigma(P) = \{u_1, u_2, \ldots, u_p\}$, $\sigma(Q) = \{v_1, v_2, \ldots, v_q\}$ sus respectivas secuencias. Un acoplamiento o emparejamiento L entre P y Q es una secuencia:

$$(u_{a_1}, v_{b_1}), (u_{a_2}, v_{b_2}), \ldots, (u_{a_m}, v_{b_m});$$

de distintos pares de $\sigma(P) \times \sigma(Q)$ tales que $a_1 = 1, b_1 = 1, a_m = p, b_m = q$ y para $i = 1, \ldots, m$ $a_{i+1} = a_i \vee a_{i+1} = a_i + 1 \wedge b_{i+1} = b_i \vee b_{i+1} = b_i + 1$, por lo que un acoplamiento respeta el orden de los puntos en P y Q.

La longitud $\|L\|$ del acoplamiento L es definida por:

$$\|L\| = \max_{i=1,\ldots,n} d(a_i, b_i).$$

La distancia discreta de Fréchet entre las cadenas poligonales P y Q es definida por:

$$d_{Fd}(P,Q) = \text{mín}\{\|L\|\}$$

con L un acoplamiento entre P y Q.

Si las dos curvas parametrizadas resultan ser cadenas poligonales, tiene sentido calcular su distancia de Fréchet mediante su distancia discreta.

La distancia discreta de Fréchet resulta ser una métrica en el conjunto constituido por las cadenas poligonales.

Intuitivamente podemos asumir que la cadena lateral de un aminoácido constituye una cadena poligonal, y de esta manera nos queda definida de forma natural una medida de

disimilitud $d_{Fd}(A, B)$ entre dos aminoácidos A y B.

RMSD: *Root-Mean Squared Deviation*

Un importante problema en la determinación y modelación de una estructura proteica es la superposición y el alineamiento con otras estructuras proteicas en un espacio tridimensional.

Un enfoque convencional para superponer un grupo de estructuras es trasladar y rotar tales estructuras de manera tal que las media aritmética de las diferencias de coordenadas de los correspondientes átomos de las estructuras, llamadas las desviaciones en media cuadrada (*Root-Mean Square Deviation*), sean minimizadas. O sea, la mejor superposición de las estructuras es obtenida cuando se alcanza la mínima desviación en media cuadrada. Podemos denotar este valor como $RMSD$ y es utilizado como una medida de similitud entre las estructuras, ver [23] para más información.

A continuación adaptaremos el método general $RMSD$ utilizado en estructuras proteicas para las cadenas laterales de los aminoácidos.

Sean $x = (x_{i,1}, x_{i,2}, x_{i,3})$ y $y = (y_{i,1}, y_{i,2}, y_{i,3})$ $i = 1, \ldots, n$ dos conjuntos de coordenadas de los átomos seleccionados para ser alineados de dos estructuras dadas respectivamente. Supongamos que x, y han sido trasladados tal que sus centros geométricos se encuentran en el origen de coordenadas. Sea Q una matriz de rotación.

El $RMSD$ entre las dos estructuras viene dada por:

$$RMSD(x, y) = \sqrt{\min_{Q} \frac{\sum_1^n (x_{i,1} - y'_{i,1})^2 + (x_{i,2} - y'_{i,2})^2 + (x_{i,3} - y'_{i,3})^2}{n}},$$

donde: $y'_i = Qy_i$.

Dadas dos proteínas A y B representadas por sus matrices de coordenadas X y Y; la superposición óptima entre estas dos estructuras en función de su valor de RMSD, puede ser determinada con los siguientes pasos según [23]:

1- Las dos estructuras necesitan ser trasladadas tal que sus centros geométricos se localicen en el mismo sitio (e.g., el origen de coordenadas)

   Sean $X = \{x_{ij}\}$, $Y = \{y_{ij}\}$ $i = 1, \ldots, n, j = 1, 2, 3$

   Los centros geométricos se pueden calcular mediante la fórmula:

$$x_c(j) = \sum_{i=1}^{n} \frac{x_{ij}}{n},$$

$$y_c(j) = \sum_{i=1}^{n} \frac{y_{ij}}{n}.$$

Sean $X' = \{x'_{ij}\}$, $Y' = \{y'_{ij}\}$ con $x'_{ij} = x_{ij} - x_c(j)$ y $y'_{ij} = y_{ij} - y_c(j)$ las matrices de las coordenadas de las estructuras trasladadas

2- Debemos determinar una matriz de rotación Q tal que:

$$\min_{Q} \|X' - Y'Q\|_F,$$

donde $\|\cdot\|_F$ denota la norma de Frobenius.

Sea $C = Y'^T X'$, y la descomposición en valores singulares de $C = U\Sigma V^T$.

La Q_{opt} óptima queda determinada por $Q_{opt} = UV^T$, por lo que la RMSD entre X y Y es:

$$RMSD(X, Y) = \frac{\|X' - Y'Q_{opt}\|_F}{\sqrt{n}}.$$

La adaptación de este método es inmediata para los aminoácidos, ya que X, Y serían las matrices de coordenadas de dos aminoácidos.

2.2. Métrica matricial para datos espaciales

Debido la naturaleza de los datos, es evidente que las medidas usuales para hallar similitudes o disimilitudes en la muestra no utilizan exactamente la estructura matricial de los aminoácidos. Es nuestro interés utilizar esta característica intrínseca de los aminoácidos para hacer los clusters.

Primero recordemos una característica de los aminoácidos que nos van a permitir preparar los datos en bruto para su uso ulterior:

- El *backbone* en todo aminoácido presenta una configuración casi invariable en cuanto que la distancia entre cada átomo presenta un comportamiento muy uniforme y los ángulos formados por la unión de los enlaces en cada átomo son muy similares también.

Esta propiedad química nos permite realizar una "traslación" o superposición de manera que coincidan todos los *backbone* de la muestra a uno específico (digamos el primer elemento de la muestra), así obtenemos un espectro de cadenas laterales donde al definir una

distancia podemos determinar la similitud entre dos de dichas cadenas. Dicha traslación se realiza de la siguiente manera:

Supongamos el aminoácido A como el pivote de la muestra (el elemento cuyo *backbone* haremos coincidir con todos los restantes *backbones*), sea B otro aminoácido.

Primero realizamos una traslación de manera que coincida el primer átomo del *backbone*, $B^1(i,j) = B(i,j) + h(j)$, $j = 1,2,3$ donde $h(j) = A(1,j) - B(1,j)$ $i = 1,\ldots,m$, y m me denota el número átomos del tipo de áminoacido de la muestra de A y B, o sea la cantidad de filas de su matriz de coordenadas.

A continuación repetimos este proceso con el segundo átomo, pero acarreando la transformación anterior, es decir:

$$\begin{aligned} B^2(i,j) &= B^1(i,j) + h(j), j = 1,2,3, i = 2,\ldots,m \\ h(j) &= A(2,j) - B^1(2,j). \end{aligned}$$

Y así sucesivamente hasta hacer coincidir todos los átomos del *backbone*.

La secuencia de transformaciones sería:

$$\begin{aligned} B^k(i,j) &= B^{k-1}(i,j) + h(j), j = 1,2,3, i = k,\ldots,m \\ h(j) &= A(k,j) - B^{k-1}(k,j), k = 1,2,3,4. \end{aligned}$$

En la Figura 2.2 presentamos las asparaginas de la Figura 2.1 después de ser trasladadas.

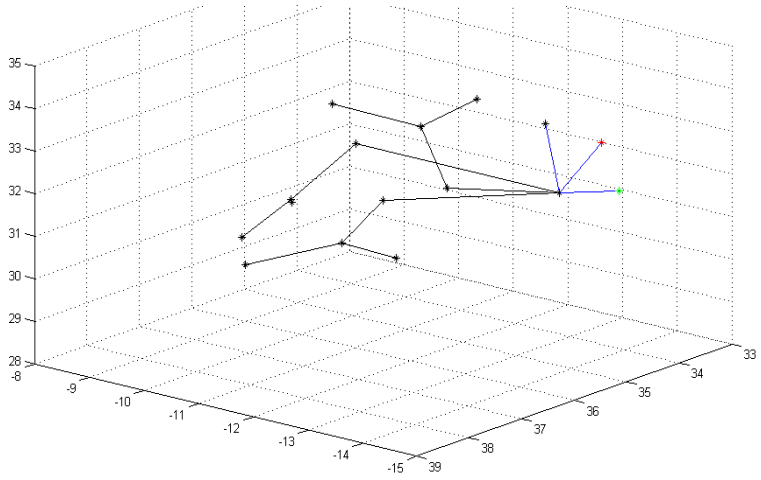


Figura 2.2: Asparaginas después de realizada la traslación

Después de realizar este tratamiento a los elementos de una muestra procedemos a calcular su medida de disimilitud. La variante con la que trabajamos fue la distancia inducida de la norma de Frobenius, para determinar cuan similares son dos datos, o sea:

***Definición* 2.2.1** *Sea A una matriz, entonces la norma de Frobenius se define como:*

$$\|A\|_F = (\sum_i \sum_j |a_{ij}|^2)^{1/2} = [\mathrm{tr}(A^t \cdot A)]^{1/2}.$$

Y la distancia asociada para dos matrices A y B es

$$d(A, B) = \|A - B\|_F.$$

En nuestro caso A, B son dos aminoácidos del mismo tipo, en el espacio $\mathbb{R}^{m\times 3}$.

Capítulo 3

Aplicaciones

Los datos de nuestro estudio son extraídos de la *Protein Data Bank*, dichos datos reciben la transformación descrita en el capítulo anterior para su posterior análisis mediante algoritmos de conglomerados, los softwares que se utilizaron para los distintos ejemplos fueron el STATISTICA 8 para el método de Ward y sus correspondientes dendrogramas, y MATLAB para los algoritmos de *Single Link* y sus dendrogramas, y para el método de k-means, el cual fue implementado en dicha plataforma, debido a la ausencia de un software que aplique este método para datos matriciales. Además se implementaron las adaptaciones de las distancias tratadas en el capítulo anterior en aras de establecer comparaciones con la distancia de Frobenius.

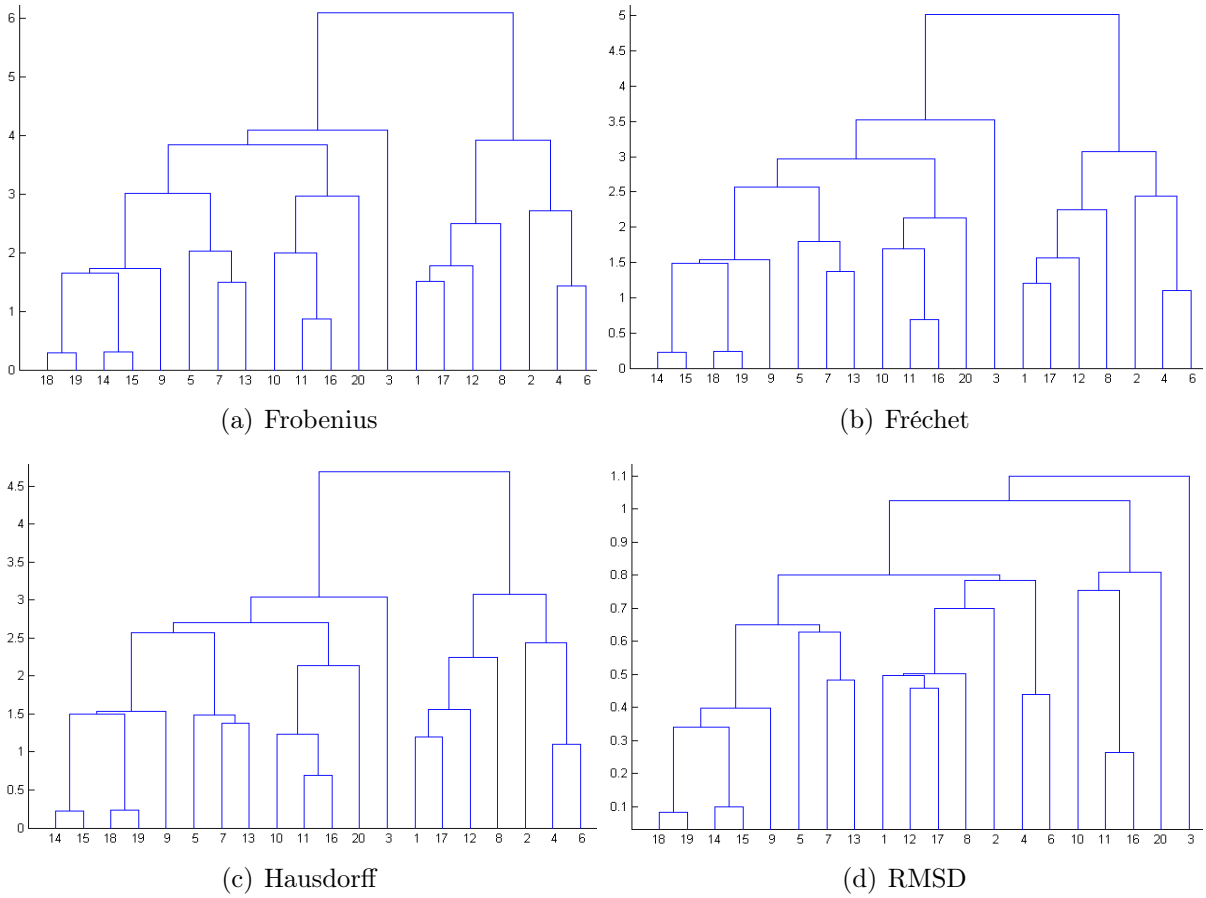
3.1. Descripción del estudio

A continuación se realiza el estudio de tres aminoácidos específicos, la cisteína (de 6 átomos), la asparagina (de 8 átomos) y la lisina (de 9 átomos), las muestras son de 1010, 1002, y 1011 datos respectivamente. Inicialmente nos concentraremos en submuestras de las mismas para establecer comparaciones en cuanto a algoritmos y entre las distintas distancias mencionadas en nuestra investigación.

3.2. Comparación de distancias y métodos

Para el primer estudio tomamos una muestra de 20 cisteínas y le aplicamos el método de *Single Link* con cada una de las distancias propuestas anteriormente.

En la Figura 3.1 se puede observar que en todos los casos, excepto en RMSD, los clusters tienen estructuras muy similares. Las diferencias yacen en cuanto al orden de agrupamiento, pero en general los pares de elementos que se agrupan inicialmente coinciden en gran medida y en general los integrantes de los clusters superiores son los mismos, siendo las distancias de Fréchet y Frobenius las dos distancias con resultados más similares.

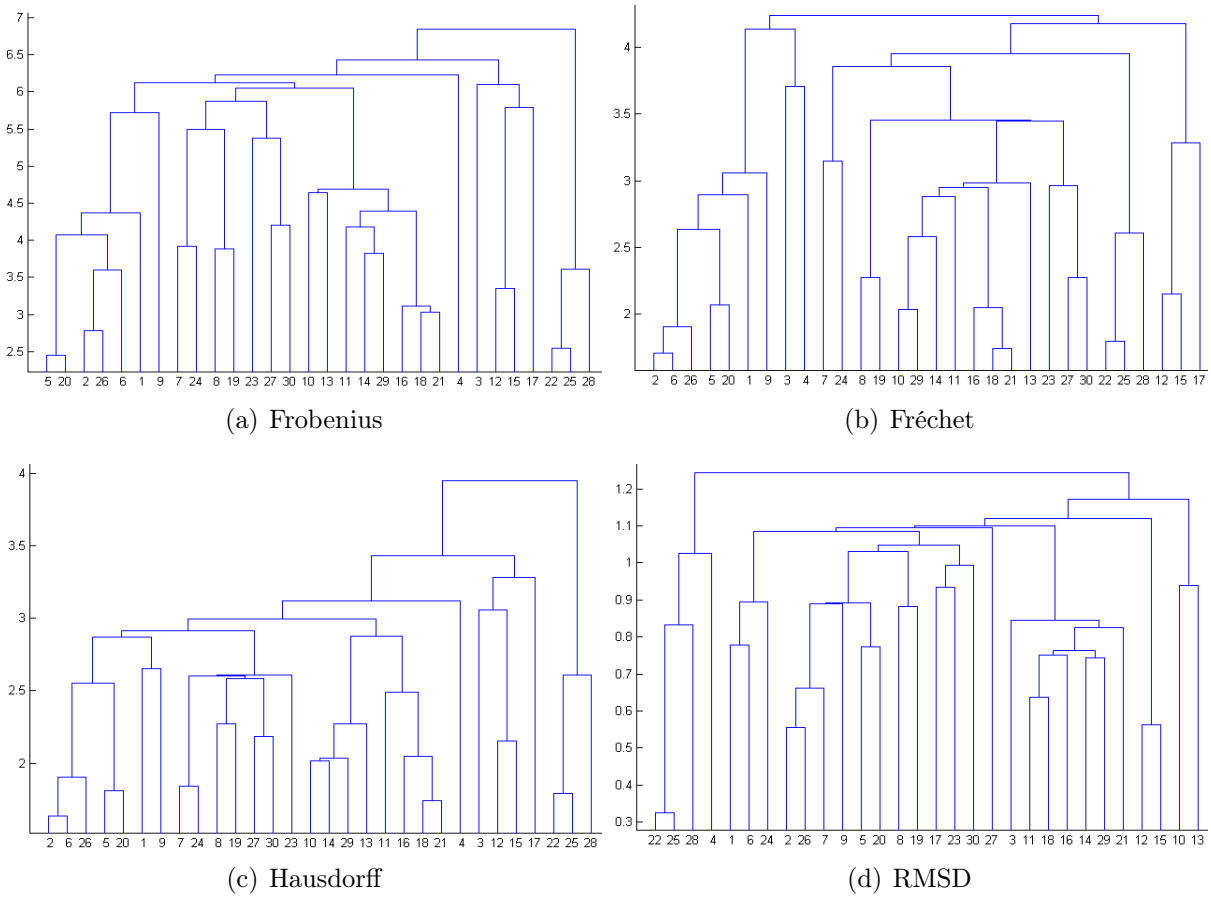


(a) Frobenius (b) Fréchet

(c) Hausdorff (d) RMSD

Figura 3.1: Dendrograma utilizando *Single Link* para 20 cisteínas.

A continuación, en la Figura 3.2, aumentamos el tamaño de la muestra hasta 30 y repetimos el procedimiento, para la asparagina.



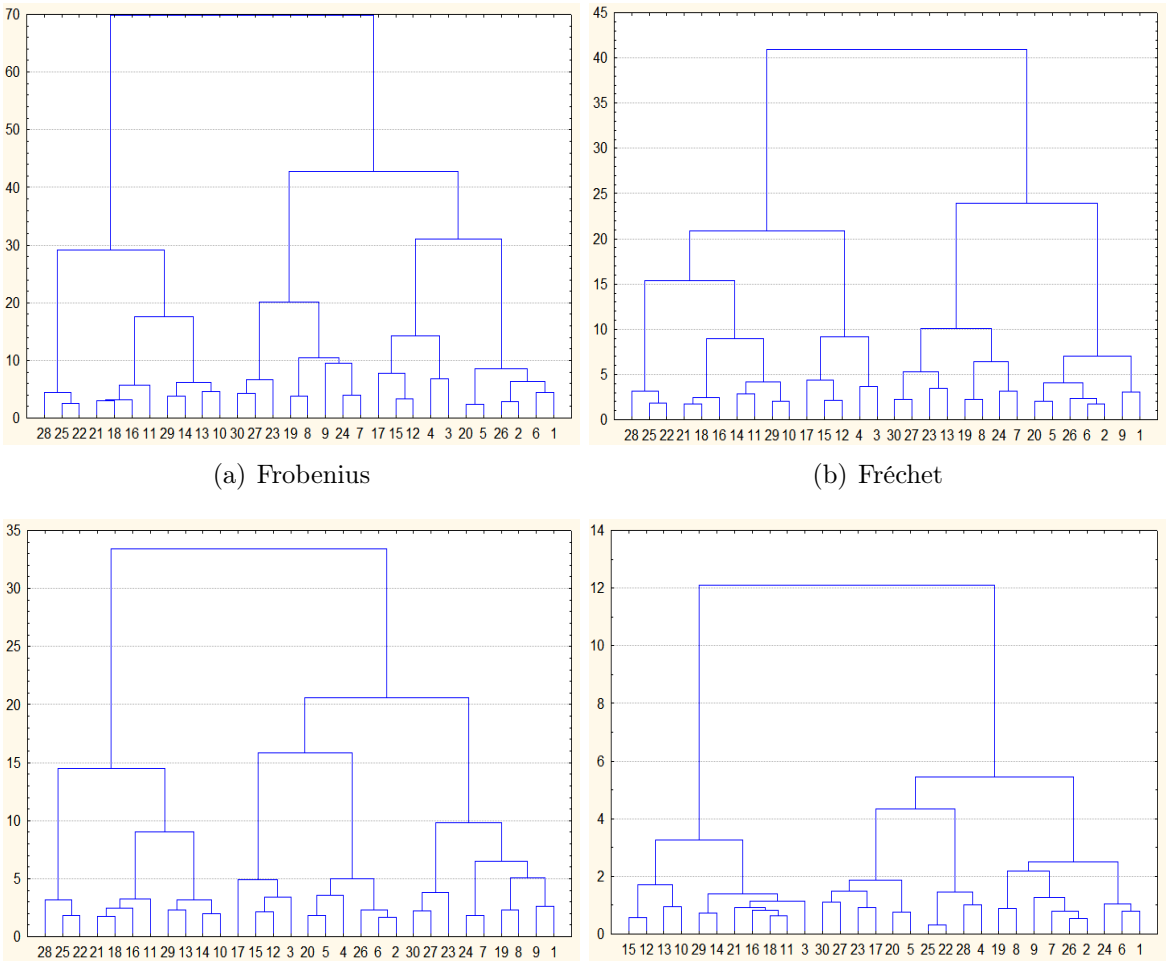
(a) Frobenius (b) Fréchet

(c) Hausdorff (d) RMSD

Figura 3.2: Dendrograma utilizando *Single Link* para 30 asparaginas.

Una vez más existen grandes analogías en la formación de los grupos con Frobenius, Fréchet y Hausdorff, se aprecia aún más las diferencias de estas tres respecto al RMSD. Esto se debe al aumento del tamaño de la muestra y a al incremento de la complejidad de la estructura (de 6 átomos en la cisteína, a 8 átomos en la asparagina).

En la Figura 3.3 se repite el paso anterior, pero utilizando el método de Ward.



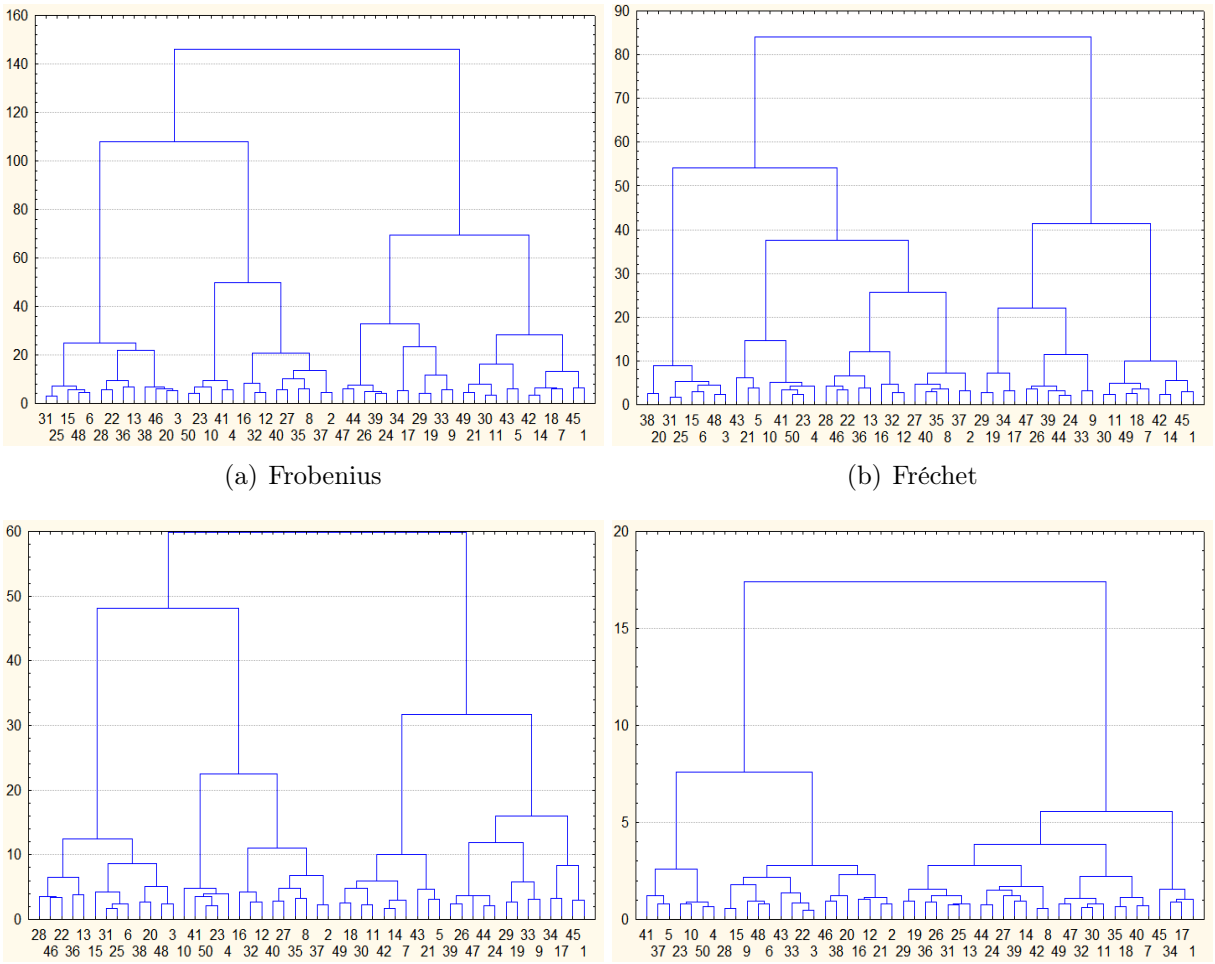
(a) Frobenius (b) Fréchet

(c) Hausdorff (d) RMSD

Figura 3.3: Dendrograma utilizando el método de Ward para 30 asparaginas.

Se observa gran similitud entre Frobenius y Fréchet, un gran número de coincidencias en los integrantes de los clusters para un corte de 7.9 en Frobenius y de 4.34 en Fréchet. Es importante señalar que la distancia de Hausdorff también guarda relación con estas dos. Una vez más la RMSD difiere en gran parte con las otras tres distancias.

Por última ocasión repetimos el proceso ahora con una muestra de 50 elementos y una estructura aún más compleja, la lisina con 9 átomos. Los resultados se pueden ver en la Figura 3.4



(a) Frobenius (b) Fréchet

(c) Hausdorff (d) RMSD

Figura 3.4: Dendrograma utilizando el método de Ward para 50 lisinas.

Aunque las diferencias son un poco más acentuadas, aún tenemos analogías en las tres primeras distancias, las diferencias se hacen incluso más notables respecto a la RMSD, por lo que podemos asumir que el RMSD (originalmente utilizada para determinar alineamiento y superposición con otras estructuras) no es la opción más adecuada para nuestro enfoque.

De las restantes tres distancias, escogeremos la distancia de Frobenius para realizar nuestros experimentos, debido a la idea intuitiva detrás de la misma, a la sencillez de su implementación y a que posee el menor costo computacional entre todas las distancias

definidas.

3.3. Clúster para aminoácidos

Se toma la muestra de 1010 cisteínas y se le aplica tanto el método de *Single Link* como de Ward.

Para ambos métodos realizaremos el siguiente análisis:

1. Aplicar el algoritmo jerárquico completo.
2. Determinar los valores de cada paso de aglomeración.
3. Tomar el 30 % del valor máximo de aglomeración y realizar un corte en ese momento.
4. De los clusters creados en dicho corte, estudiar aquellos con un número representativo de integrantes.
5. Llegar a conclusiones sobre los clusters seleccionados, para escoger la cantidad de grupos en el algoritmo de k-means.

Método	Valor máximo	Valor de corte	No. clusters	Promedio de elementos por clúster	Datos sin agrupar
Single Link	3,09	0.93	188	3.46	358
Ward	1024,61	307.38	6	168.3	0

Se observa que para el corte propuesto el método de Ward ya ha aglomerado todos los elementos en solamente 6 clusters, con un número bastante representativo de integrante por clúster, mientras que el método de *Single Link* aún le quedan elementos pendientes de análisis y sus clusters no poseen un número representativo de integrantes. En la Figura 3.5 se puede apreciar un dendrograma del método de Ward con su valor de corte.

Se aprecian la formación de 6 clusters bien definidos mediante el corte propuesto. El número de elementos por clúster es de 132; 173; 153; 164; 218 y 170 respectivamente, por lo que hemos obtenido un proceso de aglomeración bastante balanceado. En la Figura 3.6 se presentan los resultados de este análisis.

A continuación se le aplica el método de k-means tomando $k = 6$. Los resultados se pueden ver en la Figura 3.7, donde se obtienen 6 clusters muy bien definidos de 183; 129; 168; 152; 172 y 206 elementos respectivamente.

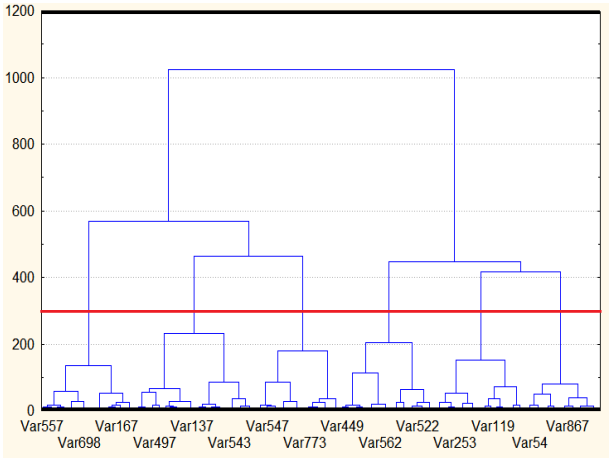


Figura 3.5: Dendrograma de 1010 cisteínas con la distancia de Frobenius y el método de Ward. Corte a 307.38 unidades.

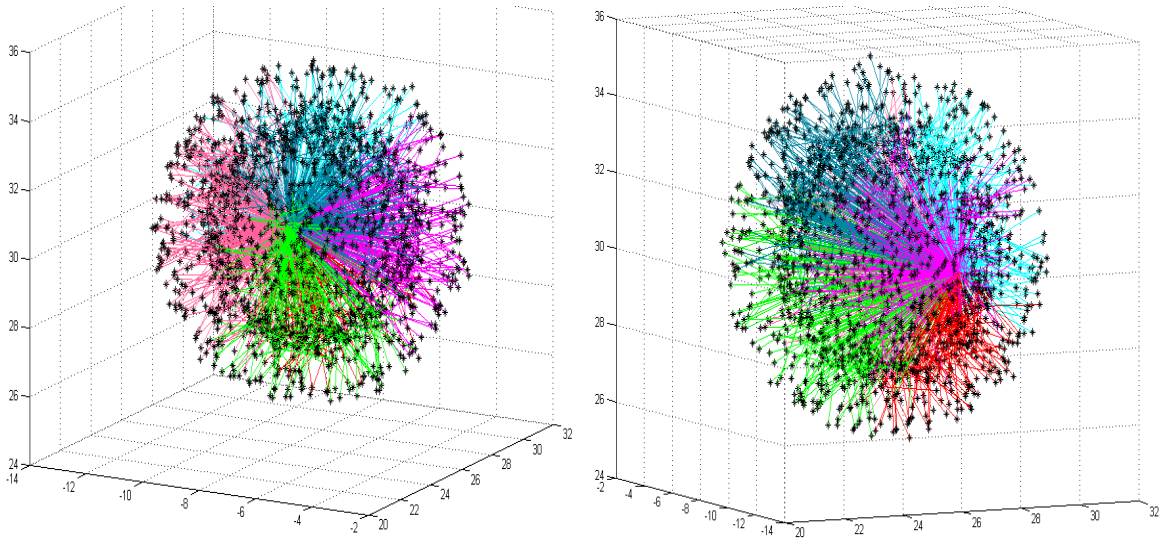


Figura 3.6: 1010 cisteínas separadas en 6 clusters método de Ward.

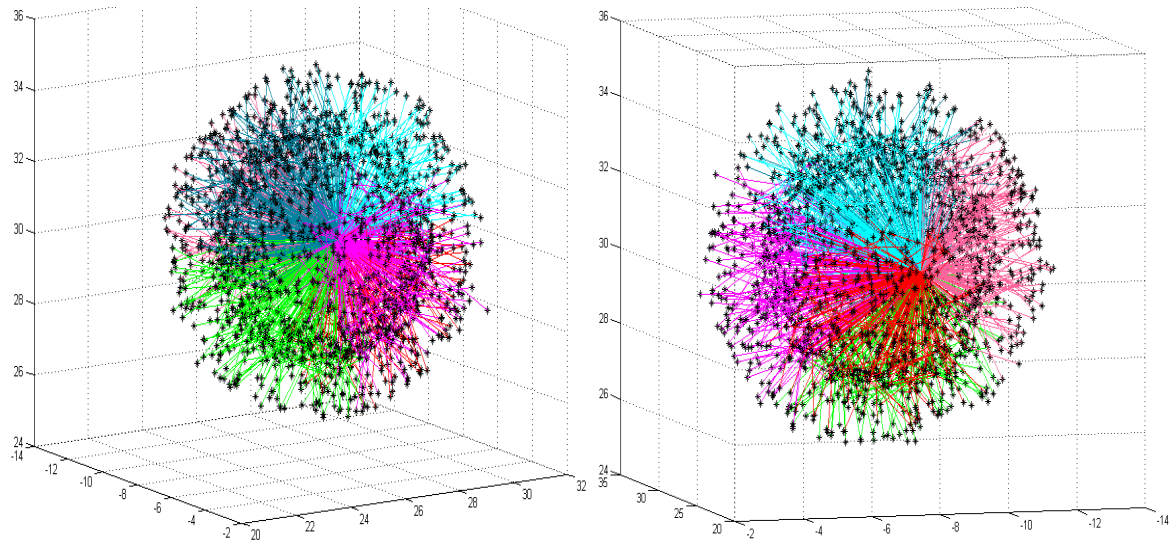


Figura 3.7: k-means para 1010 cisteínas con $k = 6$.

A continuación se le realizó el mismo estudio a la asparagina y la lisina.

Asparagina muestra 1002 datos.

Método	Valor máximo	Valor de corte	No. clusters	Promedio de elementos por clúster	Datos sin agrupar
Single Link	3,38	1.01	29	2.03	943
Ward	1285,69	385.70	7	143.14	0

Una vez más el método de Ward, ver Figura 3.8, nos brinda buenos resultados en relación con el método *Single Link*.

En la Figura 3.9 se observan 7 clusters bien definidos con 156; 103; 170; 163; 151; 135 y 124 elementos. Aplicando el método de k-means para esta misma cantidad de clusters se obtienen grupos con 140; 128; 157; 143; 131; 176 y 127 elementos que se pueden apreciar en la Figura 3.10.

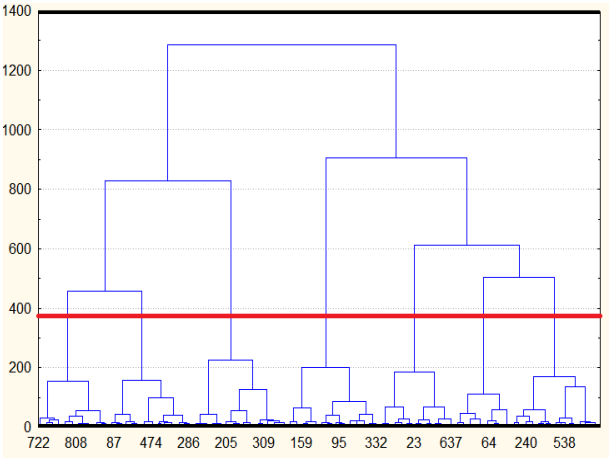


Figura 3.8: Dendrograma de 1002 asparaginas con la distancia de Frobenius y el método de Ward. Corte a 385.70 unidades.

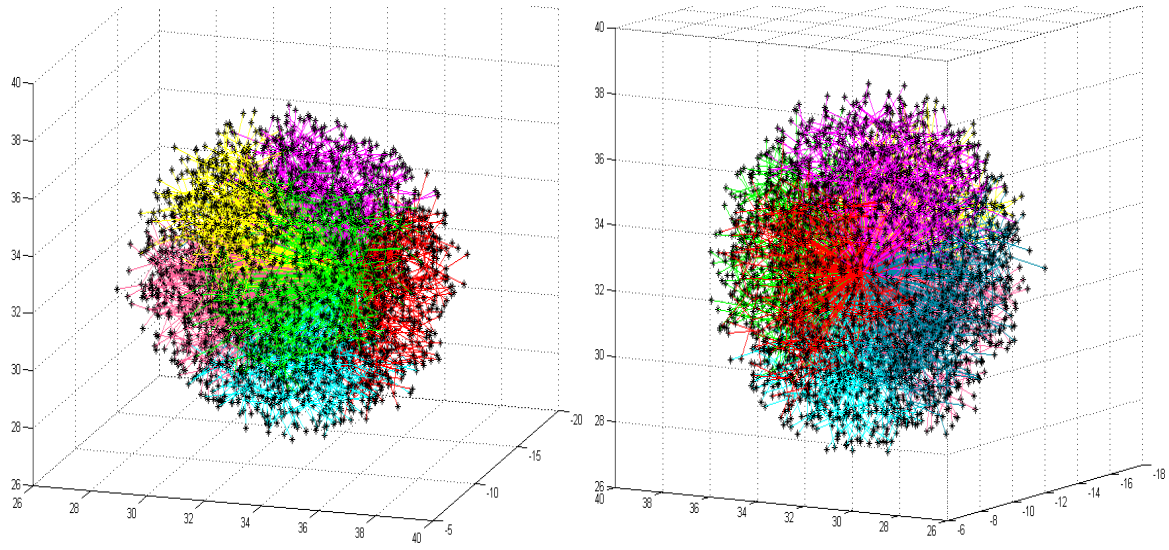


Figura 3.9: 1002 asparaginas separadas en 7 clusters método de Ward.

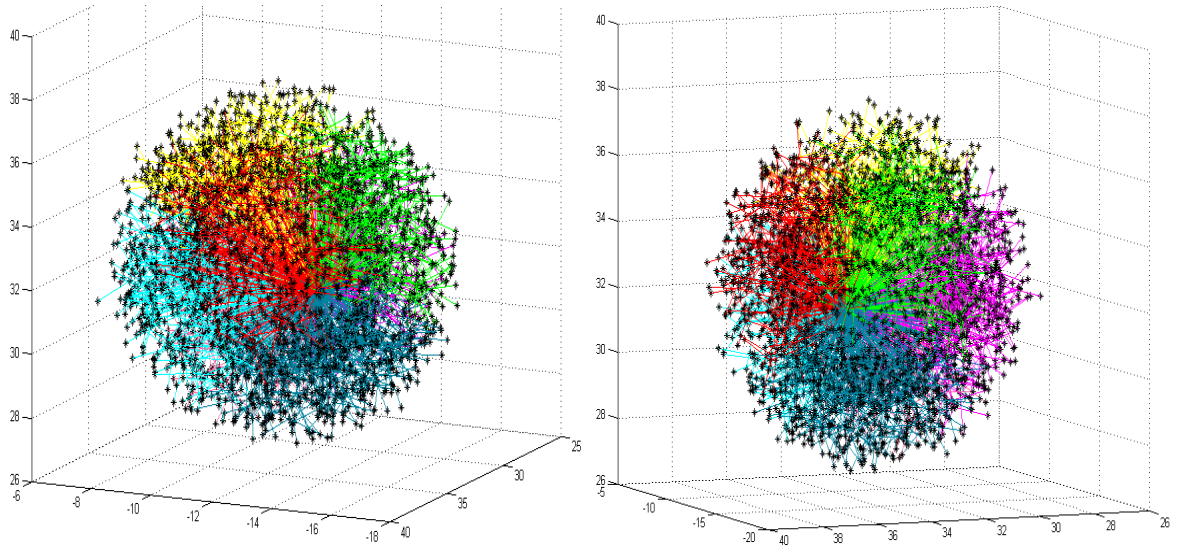


Figura 3.10: k-means para 1002 asparaginas con $k = 7$.

Lisina muestra 1011 datos.

Método	Valor máximo	Valor de corte	No. clusters	Promedio de elementos por clúster	Datos sin agrupar
Single Link	4,96	1,49	24	2	987
Ward	2053,40	616,02	6	168.5	0

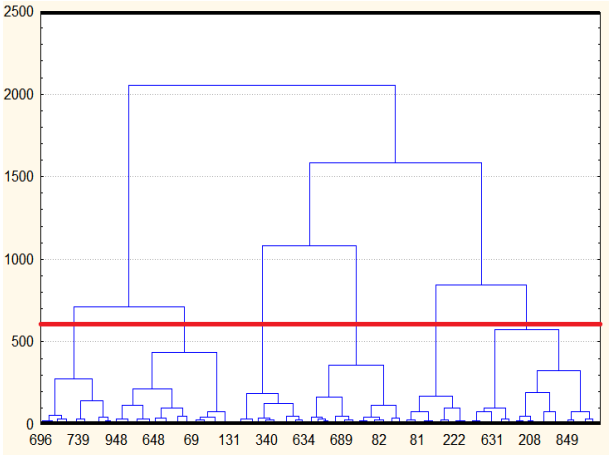


Figura 3.11: Dendrograma de 1011 lisinas con la distancia de Frobenius y el método de Ward. Corte a 616.02 unidades.

En esta ocasión los clusters están constituidos por 237; 116; 180; 214;133 y 131 integrantes respectivamente. En la Figura 3.12 se muestran los resultados de este método. Como antes, al realizar el método de k-means con el mismo número de clusters, se obtienen clusters con 179; 158; 165; 175; 204 y 130 elementos respectivamente, ver Figura 3.13.

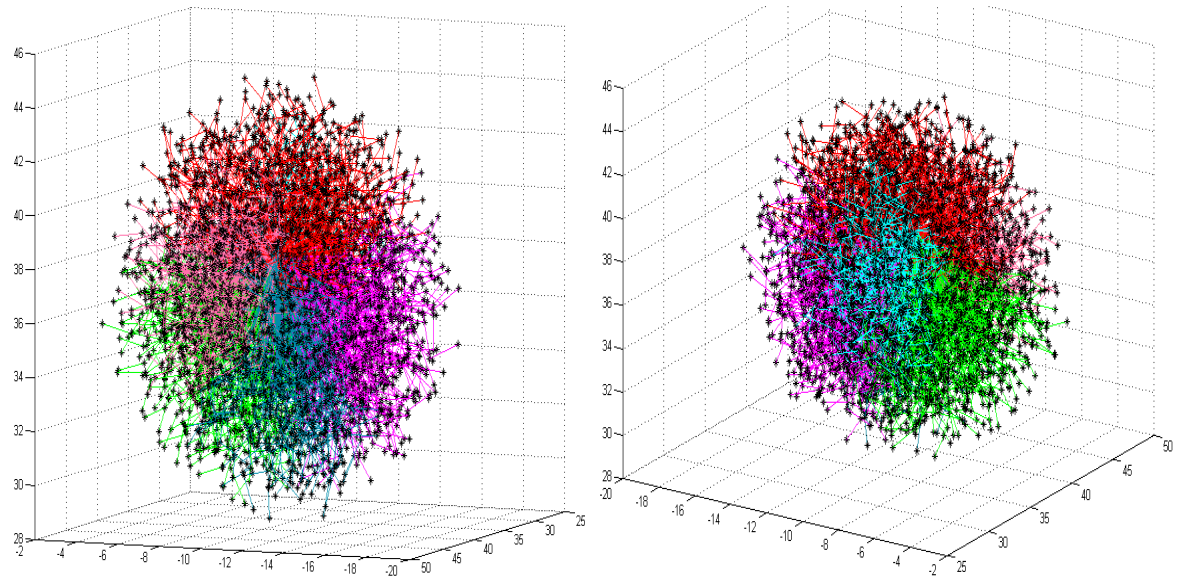


Figura 3.12: 1011 lisinas separadas en 6 clusters método de Ward.

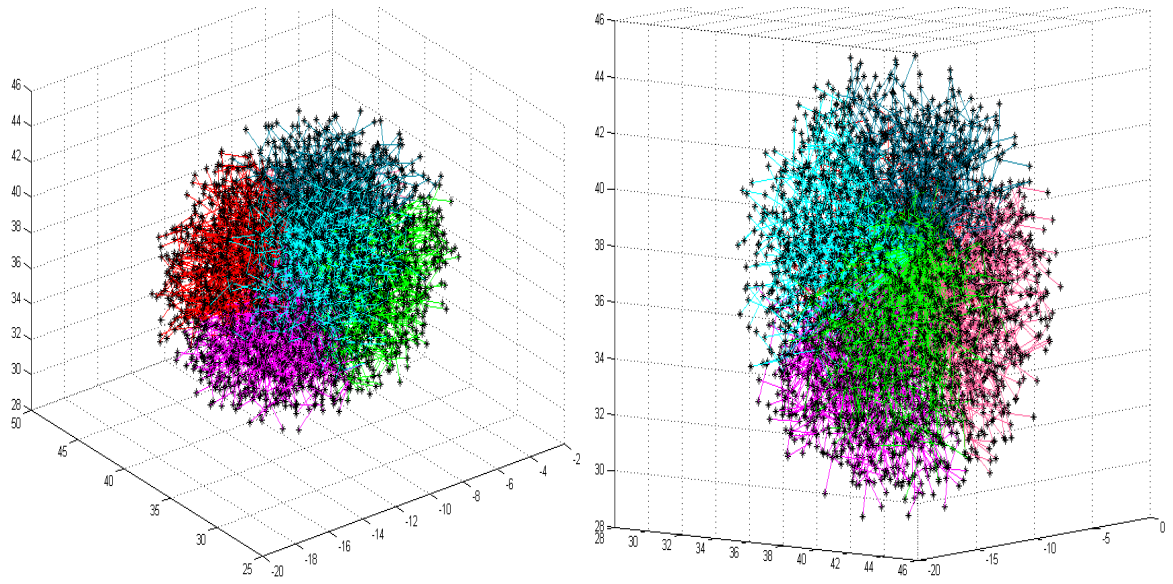


Figura 3.13: k-means para 1011 lisinas con $k = 6$.

Es importante aclarar que en todos los ejemplos el método de k-means se aplicó con una tolerancia de variación para la función de error de 0.5 unidades.

En todos los ejemplos se aprecia la funcionalidad de la distancia de Frobenius como medida de disimilitud para realizar el análisis de cluster de datos matriciales.

La creación de las librerías de conformeros se realiza a partir de los centros de los clusters resultantes en los algoritmos expuestos.

Conclusiones

A partir de la investigación realizada se han obtenido los siguientes resultados:

- Se estudiaron distancias para estructuras espaciales como: la distancia de Hausdorff, la distancia discreta de Fréchet y el RMSD, las mismas fueron adaptadas a nuestros datos para establecer comparaciones con nuestra distancia en algunos de los algoritmos mencionados, donde se apreciaron resultados muy análogos respecto a la distancia de Hausdorff y de Fréchet, teniendo la distancia de Frobenius menor costo computacional.

- Se presentó un nuevo enfoque para construir librerías de conformeros de las cadenas laterales de los aminoácidos mediante la aplicación de técnicas de análisis de clusters a partir de la estructura matricial de sus coordenadas Cartesianas. Se propuso la distancia inducida por la norma de Frobenius como medida de disimilitud para llevar a cabo dicho análisis.

- Se aplicaron los algoritmos de *Single Link*, el método de Ward y el método de k-means adaptados a datos matriciales, con ellos se evidencia la funcionalidad de la medida propuesta.

- En cada uno de los estudios realizados se obtuvieron clusters muy balanceados respecto al número de integrantes tanto en el método de Ward como en el método de k-means comprobando la calidad y funcionalidad de la distancia propuesta. El método de k-means fue implementado en MATLAB para datos matriciales, permitiendo así el uso de este algoritmo para futuras investigaciones en estudios con datos espaciales.

Recomendaciones

- Utilizar otros métodos aglomerativos más robustos como el método de grupo promedio (*Group Average Method*) y variantes del método de k-means como es el k-medoids.
- Definir métodos para elegir de número de clusters independientes del corte según el dendrograma.
- Estudiar muestras de tamaño mayor a 5000 elementos con el objetivo de crear librerías de conformeros de una mayor calidad y validar nuestros resultados a partir de bases de datos en línea.
- Emplear nuestro enfoque de análisis en otras estructuras tridimensionales que puedan ser descritas en matrices de coordenadas.

Bibliografía

[1] G. B. Coleman and H. C. Andrews. Image segmentation by clustering. *Proc. IEEE*, 67(5), 1979.

[2] Mannila H. Eiter T. Computing discrete fréchet distance. *Information Systems Department, Technical University of Vienna, Vienna, Austria.*, 94(64), 1994.

[3] M. Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884 - 1940)*, 22, 1906.

[4] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering Theory, Algorithms, and Applications*. American Statistical Association, Alexandria, Virginia, 2007.

[5] R. M. Haralick and G. L. Kelly. Pattern recognition with measurement space and spatial clustering for multiple images. *Proc. IEEE*, 57(4), 1969.

[6] Felix Hausdorff. *Grundzge der mengenlehre*. Von Veit, Leipzig, 1914.

[7] A. Hinneburg, D.A. Keim, and W. Brandt. Clustering 3d-structures of small amino acid chains for detecting dependences from their sequential context in proteins. In *Bio-Informatics and Biomedical Engineering, 2000. Proceedings. IEEE International Symposium on*, pages 43–49, 2000.

[8] L. Holm and C. Sander. Fast and simple monte carlo algorithm for side-chain optimization in proteins: application to model building by homology. *Proteins.*, 14:213–223, 1992.

[9] G.A.; Rucklidge W.J. Huttenlocher, D.P.; Klanderman. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 1993.

[10] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31, 9 1999.

[11] J. Janin, S. Wodak, M. Levitt, and B. Maigret. Conformation of amino acid side-chains in proteins. *J. Mol. Biol.*, 125:357–386, 1978.

[12] Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure-structure alignment with discrete fréchet distance. *J. Bioinformatics and Computational Biology*, 6(1):51–64, 2008.

[13] Roland L. Dunbrack Jr and Martin Karplus. Backbone-dependent rotamer library for proteins. application to side-chain prediction. *J. Mol. Biol.*, 230:543–574, 1993.

[14] Gualtieri J. A. Devney J. A. Kamgar-Parsi, B. and K. Kamgar-Parsi. Clustering with neural networks. *Biol.Cybern.*, 63, 1990.

[15] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. JohnWiley and Sons, Inc., 1990.

[16] G. Lance and W. Williams. A general theory of classificatory sorting strategies. *The Computer Journal*, 9(4), 1967.

[17] J.W. Ponder and F.M. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193:775–791, 1987.

[18] Jerod Parsons; J. Bradley Holmes; J. Maurice Rojas; Jerry Tsai; Charlie E. M. Strauss. Practical conversion from torsion space to cartesian space for *in silico* protein synthesis. *Journal of Computational Chemistry*, 26, 2005.

[19] Vivek E. P; N. Sudha. Robust hausdorff distance measure for face recognition. *Pattern Recognition*, 40, 2007.

[20] P. Tufféry, C. Etchebest, S. Hazout, and R. Lavery. A new approach to the rapid determination of protein side chain conformations. *J. Biomol. Struct. Dynam.*, 8:1267–1289, 1991.

[21] P. Tufféry, C. Etchebest, S. Hazout, and R. Lavery. A critical comparison of search algorithm applied to the optimization of protein side-chain conformations. *J. Comput. Chem.*, 14:790–798, 1993.

[22] E.J. Wood. Chemistry. an introduction to general, organic and biological chemistry. *Biochemical Education*, 13, 1985.

[23] Di Wu; Zhijun Wu. Superimposition of protein structures with dynamically weighted rmsd. *Journal of Molecular Modeling*, 16, 2010.

[24] Binhai Wylie, Tim; Zhu. Protein chain pair simplification under the discrete fréchet distance. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10, 11 2013.

[25] David C. Young. *Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems*. Wiley-Interscience, 2001.