Métodos de Kernels en secuencias para la clasificación de residuos catalíticos en sitios activos de enzimas

Nelson Hernández González

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C., Colombia
Junio, 2013

Métodos de Kernels en secuencias para la clasificación de residuos catalíticos en sitios activos de enzimas

Nelson Hernández González

Tesis presentada como requisito parcial para obtener el título de
MSc. En Ingeniería de Sistemas y Computación
Ingeniería de Sistemas

Director:
Ing. LUÍS FERNANDO NIÑO V. Ph.D.
Profesor asociado

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C., Colombia
Junio, 2013

Nota de Aceptación.

_____

_____

_____

_____

Director.

_____

Jurado.

_____

Jurado.

_____

Jurado.

Comentarios.

_____
_____
_____

Bogotá, D.C., Junio 2013

Agradecimientos

Doy gracias a Dios por permitirme culminar este proyecto.

A mis hijos Diego, Santiago y a mi esposa Andrea quienes me apoyaron en los momentos difíciles.

Al profesor Luis Fernando Niño por su continuo apoyo, constante colaboración y por su total compromiso con los estudiantes.

A todos mis compañeros del laboratorio LISI por sus aportes.

A cada uno de los profesores del departamento de Ingeniería de Sistemas que con sus enseñanzas aportaron al desarrollo del proyecto.

A las profesoras Eliana Hernández del departamento de Biología de la Universidad el Bosque y Jacqueline García del departamento de ciencias de la Universidad Santo Tomas.

Resumen

Métodos de Kernels en secuencias para la clasificación de residuos catalíticos en sitios activos de enzimas

Nelson Hernández González.
Universidad Nacional De Colombia
Director: ing. Luís Fernando Niño v. Ph.D.

Este trabajo presenta una metodología de solución al problema de clasificación de residuos catalíticos en sitios activos de enzimas. Esta metodología está basada en el aprendizaje de máquina específicamente en las máquinas de soporte vectorial (MSV); que junto a las funciones kernel permite clasificar residuos en enzimas a partir de su secuencia. El conjunto de datos utilizados fue *Catalytic Site Atlas* (CSA).

En la metodología planteada, en primer lugar encontramos la información biológica de los residuos integrada con la representación en secuencia de la enzima que lo contiene; esto por medio de las funciones kernel gaussiano y string, respectivamente. Posteriormente; el algoritmo jerárquico AGNES (*Agglomerative Nesting*) es aplicado para obtener un número de grupos inicial para el algoritmo de agrupación k-medias; obteniendo como resultado cinco grupos de enzimas. Por último, para cada grupo se desarrolló un sistema basado en MSV. La estimación del error de generalización después de validación cruzada es usada como criterio de desempeño del modelo.

.

Palabras clave:1) Máquinas de soporte vectorial 2) funciones kernel 3) sitios catalíticos

Abstract.

Kernels methods in sequences for classifying of catalytic residues in enzyme active sites.

Nelson Hernández González.
Universidad Nacional De Colombia
Director: ing. Luís Fernando Niño v. Ph.d.

This project presents a methodology to solve the problem of classification of catalytic residues in enzyme active sites. This methodology is based on machine learning and more specifically support vector machine (SVM); which together with the kernel functions allows classifying residues in enzyme with their own sequence. The dataset used during this study was Catalytic Site Atlas (CSA).

In the proposed methodology, first it is found the biologic information of the residues integrated with the sequence representation of the enzyme that contains the residue. This is done by means of the Gaussian and string kernel functions, respectively. Afterwards, the hierarchical clustering algorithm AGNES (Agglomerative Nesting) is applied in order to get a number of groups to initialize the k-means clustering algorithm, obtaining as result five groups of enzymes. Finally, for each one of the clusters, it was developed a sorting system based on SVM. The estimation of generalization error using cross validation is used as criteria of model performance.

Keywords: 1) Support Vector Machine 2) kernel functions 3) catalytic sites

Contenido

Pág.

Lista de tablas

Pág.

Lista de figuras

Pág.

Introducción

Uno de los proyectos de investigación que ha generado gran expectativa y repercusión científica es la secuenciación del genoma humano. A través de este proyecto muchos investigadores han centrado su interés en la bioinformática que se define como una ciencia que involucra el trabajo de dos grandes áreas del conocimiento, las ciencias biológicas y las ciencias de la computación y que tiene como objetivo principal el desarrollo de herramientas computacionales que permitan comprender mejor la estructura y los procesos biológicos de los seres vivos.

Los avances tecnológicos de los últimos años han permitido un crecimiento exponencial del número de estructuras proteicas determinadas experimentalmente. Actualmente existe una brecha, que cada día se va ampliando más, entre la producción de información de la secuencia de las proteínas y la determinación de su funcionalidad. Este desbalance entre estructura y función representa gran interés biológico debido a que se conoce la estructura de una proteína pero no se conoce su función o el papel biológico que desempeña. Por otro lado, determinar dicha estructura demanda un alto costo económico y de tiempo [12] [8] [9] debido a que se requiere trabajar a nivel molecular, lo que representa una extensa experimentación. Surge así la necesidad de desarrollar un conjunto de métodos computacionales que permitan realizar inferencia a partir de las secuencias de las proteínas estudiadas experimentalmente, que sean mucho más rápidos y menos costosos que la mayoría de los métodos de análisis experimental.

La Bioinformática es una de las áreas de trabajo donde las técnicas de aprendizaje de máquina (ML, por su sigla en inglés para Machine Learning), y en especial las Máquinas de Soporte Vectorial (MSV), han despertado gran interés

debido a que permiten resolver problemas biológicos usando métodos computacionales y estadísticos. Se entiende como sistemas basados en aprendizaje de máquina a aquellos capaces de optimizar un criterio de desempeño usando datos de ejemplo o experiencia pasada [15].

Uno de los mayores retos biológicos que se tiene a nivel celular es comprender los procesos regulados por proteínas, los cuales podrían ayudar a entender mejor las causas de enfermedades y la función celular; para así obtener mecanismos de intervención más eficaces en el tratamiento de enfermedades con el diseñó de drogas. Un blanco molecular específico sobre el cual puede actuar un fármaco se obtiene con la detección de un residuo específico en una secuencia de aminoácidos.

Con el análisis de la secuencia de aminoácidos de una proteína, o estructura primaria [6], se busca realizar predicciones a partir de la secuencia. Estas predicciones son útiles para obtener información sobre la región (sitio activo) de la secuencia donde se realiza la reacción química con otras proteínas [6] y también sobre la función que debe cumplir dicha proteína. Con este tipo de predicciones se pueden llevar a cabo predicciones de mayor complejidad (aspectos estructurales de mayor dimensión).

La identificación de estos sitios de unión (o de catálisis) se puede ver como un problema de clasificación binaria donde una secuencia puede tener o no un sitio activo. Recientemente se han utilizado clasificadores derivados de la teoría de aprendizaje estadístico, postulada por Vapnik y Chervonenkis en 1992, usando las Máquinas de Soporte Vectorial (MSV) [11]. Las MSV se han utilizado para resolver problemas como clasificación de proteínas, predicción funcional de una proteína, predicción de sitios activos, predicción de estructura, interacción proteína-proteína, entre otros, [8] [4], basados en propiedades fisicoquímicas y no solamente en la secuencia de las proteínas.

Los métodos de ML ofrecen una serie de clasificadores que permiten identificarlos sitios activos de una proteína desde la secuencia; en especial se centra el interés en el método Máquinas de Soporte Vectorial como clasificador que junto con las funciones kernel han sido particularmente atractivos para análisis biológicos, especialmente cuando se trata con secuencias biológicas, que son un tipo de dato muy común en bioinformática.

En este trabajo se desarrolló un modelo de clasificación a partir de uno de los métodos de aprendizaje de máquina, las (MSV), que junto con las funciones kernel [5] presentan una alternativa de solución a la clasificación de residuos catalíticos en sitios activos de enzimas a partir de la secuencia. Para resolver este problema, fue necesario determinar las funciones kernel apropiadas para datos que corresponden al tipo de secuencia de caracteres. También se propuso un método para el tratamiento de los datos que permitiera, según la función de kernel implementada, calcular la matriz de similitud entre las secuencias de proteínas. Se desarrolló un prototipo de software para el sistema de clasificación, basado en Máquinas de Soporte Vectorial (MSV).

En el primer capítulo de este documento se plantea el problema a resolver junto con los conceptos de carácter biológico y de aprendizaje maquinal necesarios para enfrentar el problema. Además, se presentan trabajos previos de identificación de sitios activos de proteínas. En el segundo capítulo se plantean y describen las funciones kernel para datos de tipo secuencia de caracteres. En el siguiente capítulo se realiza una descripción del conjunto de datos y se prueban las funciones kernel. En el cuarto capítulo se plantea la metodología desarrollada para la formulación del modelo propuesto, además, se evalúa la complejidad de dicho modelo**.** La validación del modelo y los experimentos realizados se exponen en el capítulo quinto. En el último capítulo se presentan las conclusiones de la investigación y recomendaciones sobre trabajo futuro.

1. El problema: identificación de sitios activos en secuencias de proteínas

Luego de conocer las secuencias de ADN de diferentes seres vivos, a partir de los trabajos en genómica; gran parte de las investigaciones están enfocadas en conocer la función de las proteínas, que constituyen un elemento fundamental para comprender la función de los genes. Determinar la función de una proteína a través de su estructura es un objetivo primordial para los investigadores en las ciencias de la vida y se constituye en uno de los temas de mayor interés en la biología [3].

El reconocimiento de los sitios activos es un paso importante para entender la interacción entre proteínas y conocer así los roles funcionales de cada una de ellas. Esto porque, proporcionan información útil del mecanismo de reacción entre las enzimas y sus catalizadores.

Un enfoque para detectar sitios en proteínas se basa en información de la estructura tridimensional(3D), determinada por la secuencia, en donde se considerando que algunos sitios están formados por aminoácidos que se encuentran distantes en la secuencia, pero cercanos en estructura 3D [38] [8]; esto debido a los enlaces que se forman en dicha estructura, sin embargo, surgen las preguntas: ¿Es posible detectar sitios activos considerando únicamente la secuencia de la proteína? y ¿Que tanta información se puede obtener? Estas preguntas se enmarcan en una nueva perspectiva basada únicamente en la secuencia de las proteínas sin tener presente su información estructural ni propiedades fisicoquímicas. Bajo esta perspectiva, el predecir los sitios activos de una proteína desde su secuencia primaria representa un mayor grado de dificultad,

debido a que estos sitios tienen características bioquímicas que no son apreciables desde la secuencia.

Las preguntas que se plantean conllevan a una serie de problemas que deben ser considerados:

- La representación de las secuencias de enzimas como datos de entrada a un clasificador. Teniendo presente que los clasificadores en general recurren a una representación de los datos en forma de vector; llamado vector de características.

- Determinar una transformación adecuada que traslade las secuencias de enzimas en un espacio donde sean separables linealmente.

En bioinformática un tipo común de datos son la secuencias de aminoácidos; por ejemplo las proteínas son representadas como secuencias de aminoácidos de longitud variable lo cual hace que no sea adecuada su representación como un vector de características; primero debido a la alta dimensionalidad que debería tener dicho vector, ya que está dada por la cantidad de aminoácidos de cada proteína; el segundo inconveniente que se presenta también se deriva de la dimensión de los vectores debido a que no sería posible representar todas los proteínas en un único espacio $\mathbb{R}^n$ por la longitud variable de las secuencias; tercero, la cantidad de ruido que se presenta con este tipo de representación vectorial afectaría considerablemente cualquier análisis de datos que se desee realizar; en especial al aplicar algoritmos de clasificación o predicción en secuencias de enzimas.

Por otro lado, en el espacio de las secuencias, el clasificador requerido sería un separador no lineal lo que implica extender el concepto de las MSV a clasificadores no lineales, y para ello se requiere representar los datos en un

espacio de mayor dimensión donde sea posible realizar una clasificación de dichas secuencias por medio de un separador lineal; recurriendo a una transformación que tome los elementos de entrada, secuencias en este caso, y los mapee en un espacio donde sean separables linealmente.

1.1. Conceptos biológicos

La Biología Molecular es una ciencia cuyo objetivo fundamental es la comprensión de todos aquellos procesos celulares, que contribuyen a que la información genética se transmita eficientemente de unos seres a otros, y se exprese en los nuevos individuos. El área que relaciona las proteínas con sus genes se llama proteómica, y tiene como objetivo principal resolver la pregunta ¿Qué función tiene una proteína? Algunos conceptos que se deben tener en cuenta en esta área se presentan a continuación.

1.1.1 Secuencias biológicas. Las secuencias biológicas son de tipo proteicas (secuencia de aminoácidos) ó del tipo genómicas (secuencias nucleotídicas). En las secuencias nucleotídicas cada nucleótido es un ensamble de tres componentes una base nitrogenada (adenina, timina, guanina, citosina ouracilo), un azúcar y un ácido fosfórico.

El ADN (ácido desoxirribonucleico) y el ARN (ácido ribonucleico) son moléculas poliméricas hechas de cadenas lineales de nucleótidos. La adenina, guanina, y citosina se encuentran tanto en el ADN como en ARN, mientras que la timina se encuentra sólo en el ADN y el uracilo sólo en el ARN.

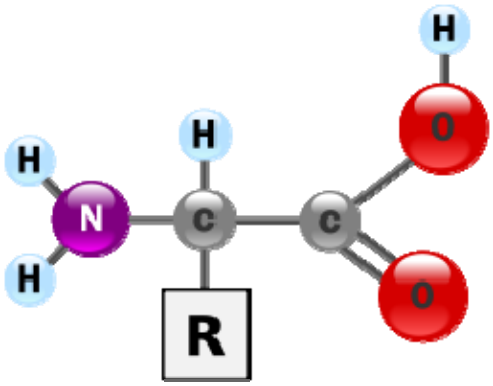
La siguiente tabla resume las secuencias biológicas.

Tabla 1. Caracteres de tipo biológico

Dato Biológico	Caracteres Biológicos
ADN	Base Nitrogenada A,C,G,T
ARN	Base Nitrogenada A,G,C,U
PROTEÍNA (aminoácidos)	A,V,F,P,M,I,L,D,E,K,R,S T,Y,H,C,N,Q,W,G

1.1.2. Aminoácidos. La estructura fundamental de cada aminoácido está dada por un átomo de carbono central, un grupo amino (NH2), un grupo carboxilo (COOH), un átomo de hidrógeno y un grupo de átomos denotados por R que se enlazan como se muestra en la figura siguiente.

Figura 1. Estructura fundamental de los aminoácidos



La diferencia entre aminoácidos está dada por el grupo de átomos R (grupo lateral) es así que existen 20 tipos diferentes de aminoácidos para construir las proteínas. La molécula que se forma por la unión de muchos aminoácidos se llama polipéptido [17].

Tipos de aminoácidos: alanina, arginina, asparagina, ácido aspártico,cisteína, glutamina, ácido glutámico, glicina, histidina, serina, tirosina,prolina, isoleucina, leucina, lisina, metionina, fenilalanina, triptófano,treonina, valina.

En la siguiente tabla se tiene el nombre completo y la abreviatura de cada aminoácido y algunas propiedades fisicoquímicas

Tabla 2. Nombre de aminoácidos y abreviatura

Abrev.		Nombre completo	Masa	Punto isoeléctrico
A	Ala	Alanina	89.09	6.11
C	Cys	Cisteína	121.16	5.05
D	Asp	Ácido aspártico	133.10	2.85
E	Glu	Ácido glutámico	147.13	3.15
F	Phe	Fenilalanina	165.19	5.49
G	Gly	Glicina	75.07	6.06
H	His	Histidina	155.16	7.60
I	Ile	Isoleucina	131.17	6.05
K	Lys	Lisina	146.19	9.60
L	Leu	Leucina	131.17	6.01

M	Met	Metionina	149.21	5.74
N	Asn	Asparagina	132.12	5.41
P	Pro	Prolina	115.13	6.30
Q	Gln	Glutamina	146.15	5.65
R	Arg	Arginina	174.20	10.76
S	Ser	Serina	105.09	5.68
T	Thr	Treonina	119.12	5.60
V	Val	Valina	117.15	6.00
W	Trp	Triptófano	204.23	5.89
Y	Tyr	Tirosina	181.19	5.64

1.2. Proteína

Las proteínas son moléculas de gran tamaño formadas por secuencias de aminoácidos, esta cadena polipeptídica determina el carácter biológico de la molécula proteica. Una pequeña variación en la secuencia puede alterar la manera de funcionar de la proteína [14].Para ensamblar los aminoácidos en proteínas cada molécula debe no solo tener una gran cantidad de aminoácidos sino también suficiente cantidad de cada tipo.

1.2.1. Estructura de las proteínas.

- Estructura primaria.

La secuencia lineal de aminoácidos, dada por la información hereditaria contenida en la célula, se conoce como estructura primaria de la proteína. Por ejemplo la proteína 1A0I de PDB (Protein Data Bank) tiene secuencia.

VNIKTNPFKAVSFVESAIKKALDNAGYLIAEIKYDGVRGNICVDNTANSYWLSRVSKTIPALEHLNGFDVRWKRL
LNDDRCFYKDGFMLDGELMVKGVDFNTGSGLLRTKWTDTKNQEFHEELFVEPIRKKDKVPFKLHTGHLHIKLY
AILPLHIVESGEDCDVMTLLMQEHVKNMLPLLQEYFPEIEWQAAESYEVYDMVELQQLYEQKRAEGHEGLIVKD
PMCIYKRGKKSGWWKMKPENEADGIIQGLVWGTKGLANEGKVIGFEVLLESGRLVNATNISRALMDEFTETVK
EATLSQWGFFSPYGIGDNDACTINPYDGWACQISYMEETPDGSLRHPSFVMF.

- Estructura Secundaria.

A medida que la secuencia de aminoácidos se ensamblan comienza a ocurrir interacciones entre ellos; se forman puentes entre el hidrógeno del grupo amino de un aminoácido y el oxígeno de otro aminoácido. Dos estructuras resultan de tales puentes una llamada hélice alfa y la otra lámina plegada beta.

- Estructura Terciaria.

Esta estructura se refiere a la interacción espacial entre los diferentes aminoácidos y la cual está relacionada con la naturaleza de los grupos R de aminoácidos individuales que actúan sobre la secuencia. La estructura tridimensional que resulta de estas interacciones entre los grupos R se denomina estructura terciaria o 3D.

1.2.2. Enzimas. Se llaman enzimas a las sustancias de naturaleza proteica que aumentan la velocidad de reacciones químicas (catalizan).Esto es, actúan facilitando las transformaciones químicas; acelerando considerablemente las reacciones y disminuyendo la energía de activación que muchas reacciones requieren.

Las enzimas actúan sobre unas moléculas denominadas sustratos, las cuales se convierten en diferentes moléculas, y productos. Una forma general de denominar a las enzimas es añadir el sufijo "*asa*" al nombre del sustrato. Así, se clasifican en 6 categorías principales dependiendo de la reacción que catalice la enzima.

- Transferasas (transferencia de grupos funcionales).
- Ligasas:
- Hidrolasas.
- Oxirreductasas:
- Isomerasas
- Liasas.

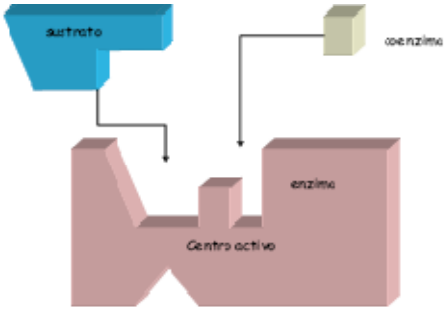
Las enzimas suelen ser muy específicas tanto del tipo de reacción que catalizan como del sustrato involucrado en la reacción.

Las moléculas del sustrato se unen a un sitio particular en la superficie de la enzima, denominado sitio activo, donde tiene lugar la catálisis. La estructura tridimensional de este sitio activo, es lo que determina la especificidad de las enzimas. El acoplamiento es tal que el sustrato se adapta al centro activo o catalítico de una enzima como una llave a una cerradura [47].

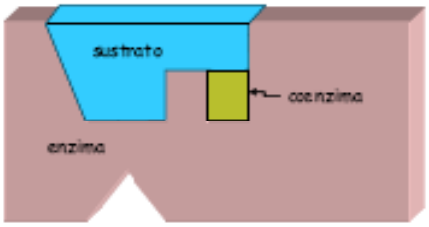
Mecanismo de acción de las enzimas. La figura 2 muestra el mecanismo de reacción de las enzimas, el cual se puede resumir en los siguientes pasos:

Figura 2. Mecanismo de reacción de las enzimas

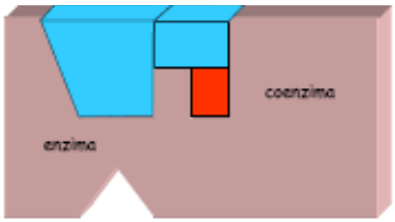
- Se forma un complejo: Enzima+substrato como se aprecia en la figura.



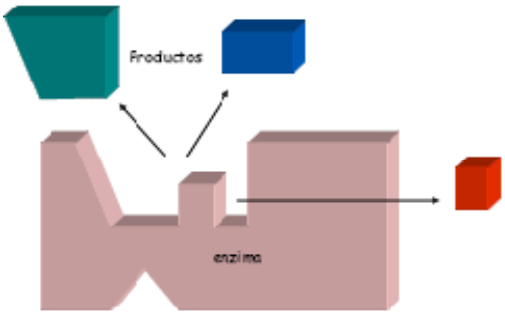
- El sustrato se unen al centro activo de la enzima.



- Los restos de los aminoácidos que configuran el centro activo catalizan el proceso.



- Los productos de la reacción se separan del centro activo y la enzima se recupera intacta.



1.2.3. Sitio activo. Al conjunto de aminoácidos que están directamente relacionados en la unión enzima-sustrato y en las posteriores reacciones químicas que se desarrollan se llama centro activo o sitio activo. Sólo una pequeña parte de la proteína forma parte del centro activo.

Un sitio activo está muy bien definido y contiene:

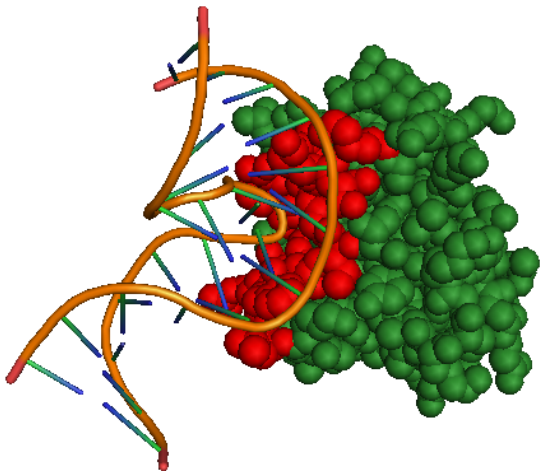
- Residuos catalíticos (necesarios para llevar a cabo la catálisis)
- Residuos de unión (aportan soporte para unir el enzima a su sustrato y no a otro)
- Residuos estructurales (que estabilizan la estructura tridimensional necesaria para el proceso catalítico).

1.2.4. Residuos catalíticos. Se pueden definir las siguientes reglas para clasificar los residuos de sitios activos como catalíticos:

- Están directamente involucrados en el mecanismo catalítico
- Ejercen un efecto en otro residuo o molécula de agua la cual está directamente involucrada en el mecanismo catalítico.
- Ejercen efecto en un sustrato o cofactor (moléculas no proteicas) que ayudan a la catálisis.
- Estabilización de un estado intermedio de transición.

La siguiente figura muestra la estructura 3D de la proteína 1T0K_B, donde se aprecia en color rojo el sitio activo o de unión de la misma. La forma en que se pliega la proteína depende de los enlaces entre aminoácidos.

Figura 3. Sitio activo de la proteína 1T0K_B



Al considerar las proteínas en su secuencia primaria el sitio activo estaría conformado por aminoácidos que no son contiguos en la secuencia, como se aprecia a continuación.

```
1T0K_B
SINQKLALVIKSGKYTLGYKSTVKSLRQGKSKLIIIAANTPVLRKSELEYYAMLSKTKV
YYQGGNNELGTAVGKLFRVGVVSILEAGDSDILTTLA
00000000000000001111100100000000000000011001000000000000000
0000001000000000111110000000000000000
```

1.3. Aprendizaje de máquina

Se entiende por aprendizaje de máquina (ML por sus iníciales en inglés) al conjunto de algoritmos y técnicas que tienen como objetivo crear sistemas capaces de optimizar un criterio de desempeño usando datos de ejemplo o experiencia pasada [15].

Algunos ejemplos de aplicación de las técnicas de aprendizaje de máquina se presentan a continuación.

1.3.1. Clasificación y agrupamiento. Las técnicas de clasificación y regresión se enmarcan en el aprendizaje supervisado, cuyo objetivo es predecir una característica y, llamada etiqueta, como función de un conjunto de características x o vector de entrada [7].

Para aplicarlas técnicas de clasificación la etiqueta y toma valor en un conjunto llamado de clases, el cual contiene la clase positiva y la clase negativa, que generalmente se representa como $\{+1, -1\}$ respectivamente.

El aprendizaje no supervisado se caracteriza por que los datos no cuentan con etiquetas, es decir, solo se tiene como entrada el vector de características x. El objetivo es agrupar los datos de forma natural; en este tipo de aprendizaje se enmarca el agrupamiento.

Las técnicas o algoritmos en la de conformación de grupos (conglomerados) se dividen en algoritmos de tipo jerárquico y algoritmos de tipo particional

Un método jerárquico aglomerativo inicia considerando cada observación como un grupo y en sucesivos pasos se van uniendo, según la similitud, hasta que finalmente todos los datos están en un único conglomerado, un algoritmo usado es AGNES (*Agglomerative Nesting*) [23].Un método jerárquico divisivo, por el contrario, parte de un gran conglomerado y en pasos sucesivos se va dividiendo hasta que cada dato resulte agrupado en un conglomerado; un algoritmo de este tipo es DIANA (*Divisive Analysis*) [23].

En un método particional o no jerárquico se divide el conjunto de datos en un número fijo de k conglomerados, definido inicialmente. Los métodos de análisis de conglomerados no jerárquico más conocidos son k-*medias* [22] y el k-*medoide* y

otros dos algoritmos derivados como PAM, por las iníciales en inglés de *Partitioning Around Medoids* [23].

1.3.2. Clasificador lineal.

La clasificación binaria es frecuentemente realizada por medio de una función $f: \mathrm{X} \subseteq \mathbb{R}^n \to \mathbb{R}$. . Así; la entrada $\mathrm{x} = (x_1, x_2, x_3, \ldots x_n)$ es asignada a la clase positiva si $f(\mathrm{x}) \geq 0$ si no, es asignada a la clase negativa.

Consideramos para esto la función lineal.

$$\begin{aligned} f(\mathrm{x}) &= \langle \mathrm{x} \cdot \mathrm{w} \rangle + b \\ &= \sum_{i=1}^{n} x_i w_i + b \end{aligned}$$

donde $(\mathrm{w}, b) \in \mathbb{R}^n \times \mathbb{R}$ son parámetros de control de la función y la regla de decisión que determina que $\mathrm{x} = (x_1, x_2, x_3, \ldots x_n)$ esté en una o en la otra clase está dada por el signo de $f(\mathrm{x})$

La interpretación geométrica de los clasificadores lineales indica que el espacio de entrada X se divide en dos clases por medio de un hiperplano con ecuación $\langle \mathrm{w} \cdot \mathrm{x} \rangle + b = 0$.

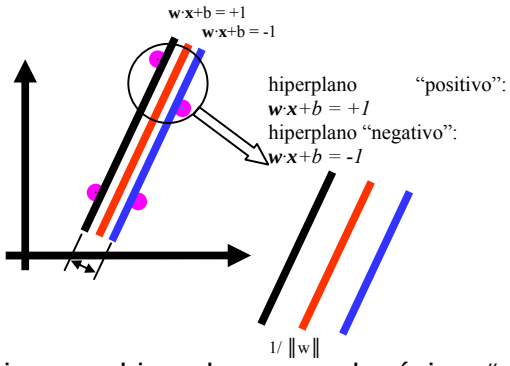
1.3.3. Máquinas de soporte vectorial. Las MSV son clasificadores derivados de la teoría de aprendizaje estadístico postulada por Vapnik y Chervonenkis. Fueron presentadas en 1992 y adquirieron fama cuando dieron resultados muy superiores a las redes neuronales en el reconocimiento de letras manuscritas, usando como entrada pixeles de una imagen.

La interpretación geométrica de las MSV se puede ver de la siguiente manera. Cada elemento del espacio de características es representado por un punto de n componentes previamente definidas. Es decir, hay l observaciones y cada una consiste en un par de datos (x_i, y_i) donde $\mathrm{x}_i \in \mathbb{R}^n, i = 1, \ldots l$ es un vector y $y_i \in \{+1, -1\}$ es la etiqueta.

Supóngase que se tiene un hiperplano que separa las muestras positivas (+1) de las negativas (-1). Los puntos x_i que están en el hiperplano satisfacen

$$\mathrm{w} \cdot \mathrm{x} + b = 0$$

Figura 4. Interpretación geométrica de las MSV



Se pretende determinar un hiperplano con el máximo "margen", el cual se puede obtener al resolver el siguiente problema de optimización.

Encontrar $\mathrm{w} \in \mathbb{R}^n$,y, $b \in \mathbb{R}$ tal que

$$\text{Maximice} : \frac{1}{\|\mathrm{w}\|}$$

$$\text{Sujeto a: } y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) \geq 1$$

con (x_i, y_i) perteneciente al conjunto de observaciones .También podemos definir el problema como:

Encontrar $\mathrm{w} \in \mathbb{R}^n$,y, $b \in \mathbb{R}$ tal que minimicen $\phi(\mathrm{w}) = \frac{1}{2}\langle \mathrm{w} \cdot \mathrm{w} \rangle$

Sujeto a: $y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) \geq 1$

Se define la función de Lagrange, para este problema como.

$$L_P(\mathrm{w}, b, \alpha) = \frac{1}{2}\langle \mathrm{w} \cdot \mathrm{w} \rangle - \sum_{i=1}^{l} \alpha_i \left(y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) - 1\right)$$

donde $\alpha_i \geq 0$ son los multiplicadores de Lagrange. Así, la función de Lagrange dual está dada por $L_D(\alpha) = min\ L_P(\mathrm{w}, b, \alpha)$.

El problema dual [7] se resuelve al encontrar α talque

Maximice:$L_D(\alpha)$

Sujeto a:$\alpha_i \geq 0$

El cálculo de $L_D(\alpha)$ se obtiene derivando parcialmente L_P. Así.

$$\frac{\partial L_P}{\partial \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{l} y_i \alpha_i \mathrm{x}_i = 0$$

$$\mathrm{w} = \sum_{i=1}^{l} y_i \alpha_i \mathrm{x}_i$$

Además,

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0$$

Al remplazar los anteriores resultados y considerando que

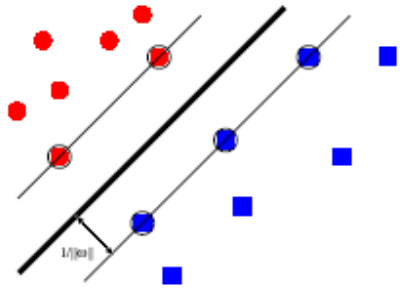
$$\begin{aligned} L_D(\alpha) &= L_P(\mathrm{w}, b, \alpha) \\ &= \frac{1}{2}\langle \mathrm{w} \cdot \mathrm{w} \rangle - \sum_{i=1}^{l} \alpha_i (y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) - 1) \\ &= \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_j \langle \mathrm{x}_i \cdot \mathrm{x}_j \rangle - \sum_{i=1}^{l} \alpha_i \left(y_i \left(\sum_{j=1}^{l} y_j \alpha_j \langle \mathrm{x}_j \cdot \mathrm{x}_i \rangle + b \right) - 1\right) \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j\, \alpha_j \langle \mathrm{x}_i \cdot \mathrm{x}_j \rangle \end{aligned}$$

Considerando las condiciones complementarias de Karush-Kuhn-Tucker (KKT) [52] para el problema de optimización se tiene que

$$\alpha_i(y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) - 1) = 0$$

Al conjunto de los puntos x_i que cumplan $y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) = 1$ para $\alpha_i \neq 0$, se les denomina vectores soporte (VS), los cuales determinan los hiperplanos que se representan en la figura 5.

Figura 5. Vectores soporte



Entonces el hiperplano buscado está dado por la ecuación

$$f_{SV}(\mathrm{x}) = \sum_{\mathrm{x}_i \in SV} \alpha_i y_i \langle \mathrm{x}_i \cdot \mathrm{x} \rangle + b$$

Cuando los datos no se pueden separar linealmente se hace un cambio de espacio mediante una función que transforma los datos de manera que se pueden separar linealmente en dicho espacio. Tal función se llama función kernel, que se presenta con más detalle en el capítulo siguiente.

1.3.4. Regularización. Al plantear un modelo para una tarea de clasificación se desea que dicho modelo tenga la mejor capacidad de generalización posible, es decir, que la capacidad de predicción sobre datos diferentes a los que se utilizaron para generar el modelo sea alta. Teniendo presente que no se tenga un sobre ajuste o por el contrario subajuste del mismo.

La regularización permite entrenar modelos de tal manera que se disminuya el riesgo de sobre ajuste, por medio del control de la complejidad del modelo [15].

Para la regularización de un modelo se considera la función de error

$$E = \textit{error de datos de entrenamiento} + \lambda\ \textit{complejidad}$$

donde λ se denomina el parámetro de regularización, el cual especifica el equilibrio entre el error de entrenamiento y la complejidad del modelo. Cuanto mayor es el valor de λ mayor la penalización de la complejidad del modelo

La regularización en MSV se realiza por medio de la constante C. Como se sabe en las MSV se busca el hiperplano que cumpla.

$$\min \frac{1}{2} \|\mathrm{w}\| + C \sum_{i=1}^{l} \xi_i$$

Sujeto a

$$y_i(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

donde (x_i, y_i) son elementos del conjunto de entrenamiento; ξ_i es el error dado por la MSV; y C controla la relación entre la complejidad del MSV y el error de entrenamiento, es decir, controla la penalización por clasificar mal un elemento del conjunto de entrenamiento. Un valor alto de C hace que la MSV genere una función de predicción bastante compleja, la cual clasifica mal unos pocos

elementos del conjunto de entrenamiento y por el contrario un valor bajo de C genera una función de predicción sencilla [49]

1.3.5. Medidas de desempeño. El desempeño se mide a partir de la matriz de confusión [22] que está dada por

	PREDICCIÓN	
	+	-
+	TP	FN
-	FP	TN

donde TP es el número de verdaderos positivos(secuencias que la función de predice con un sitio activo y que verdaderamente tiene un sitio activo),FP es el número de falsos positivos(secuencias que la función de predice con un sitio activo y que verdaderamente no tiene un sitio activo), TN es el número de verdaderos negativos y FN es el número de falsos negativos.

Se define además la exactitud, la sensibilidad y la precisión como:

$$exactitud = \frac{TN + TP}{TN + TP + FP + FN}$$

$$sensibilidad = \frac{TP}{TP + FN}$$

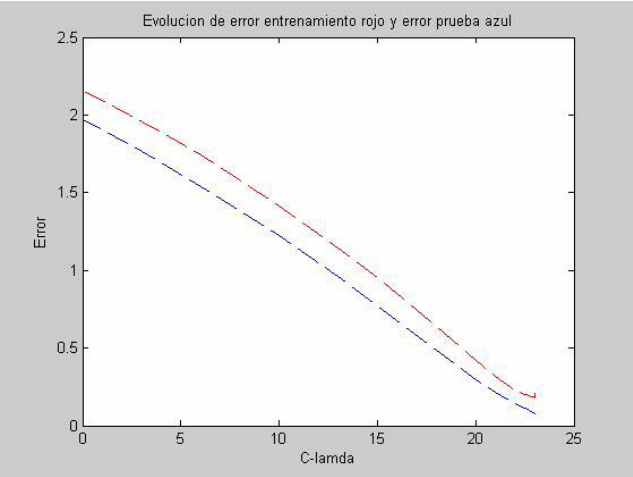
$$precision = \frac{TP}{TP + FP}$$

1.3.6. Validación cruzada con K iteraciones. La metodología utilizada para determinar los parámetros que optimizan el desempeño del modelo, se basa en la técnica conocida como validación cruzada, en la cual el conjunto de datos se divide en dos partes; el conjunto de entrenamiento (para entrenar y validar el modelo) y el conjunto llamado de prueba que es independiente de los datos de entrenamiento.

En k-iteraciones el conjunto de entrenamiento se divide a su vez en k subconjuntos; con k-1 subconjuntos se crea el modelo y con el conjunto restante se valida el modelo (conjunto de validación), este proceso se realiza k veces. El resultado del error es el promedio de los errores en las k repeticiones; este procedimiento permite ajustar los parámetros del modelo de forma tal que generen el menor error. Una vez ajustado los parámetros del modelo óptimo se mide su desempeño utilizando un segundo conjunto de secuencias llamado conjunto de prueba que es independiente del conjunto de entrenamiento.

Con el conjunto de entrenamiento se generan modelos con diferente complejidad y se evalúa el error con el conjunto de prueba de cada modelo generado. El nivel de complejidad óptimo se da donde se obtiene el "codo" en la gráfica de complejidad vs error. La figura 6 muestra un ejemplo como se determina el nivel óptimo de complejidad. En este caso la mejor complejidad se tiene en aproximadamente para un valor de C=22

Figura 6. Complejidad del modelo vs. error



1.4. Bases de datos biológicas de proteínas PDB

El Protein Data Bank (PDB) es un repositorio internacional de estructuras de proteínas. La fuente principal de información acerca de estructura 3D de macromoleculas (incluye proteínas, péptidos, virus, complejos proteína-ácidos, ácidos nucleicos, carbohidratos) es el PDB y su formato es el estándar para el intercambio de información estructural. La mayoría de estas estructuras son determinadas experimentalmente por medio de difracción de rayos X o resonancia magnética nuclear. El PDB comenzó en 1971 con 7 proteínas; actualmente (junio del 2013) contiene 90611 estructuras.

1.4.1. Atlas de sitios catalíticos (Catalytic Site Atlas). El Atlas de sitios catalíticos (CSA) es una base de datos que contiene anotaciones de sitios activos y residuos catalíticos de enzimas en estructura 3D. La versión de CSA que se utilizó contenía 6063 residuos catalíticos anotados en 1627 enzimas.

1.5. Trabajos previos en identificación de sitios activos

En [10] se propuso un nuevo algoritmo para predecir sitios activos basados en características estructurales y bioquímicas de las secuencias de amino ácidos.

El clasificador seleccionado fue Naive Bayes. Por otro lado el trabajo realizado en [8] predijo residuos catalíticos usando SMO (Sequential Minimal Optimization), que es un método para entrenar una MSV; consideraron un vector de 24 atributos estructurales de las proteínas para representar los datos de entrada. Además comparó 26 clasificadores usando la herramienta de software libre WEKA. Otro trabajo en [2] se usó una MSV para predecir interacciones proteína-proteína desde estructura primaria, considerando además las propiedades fisicoquímicas

asociadas a la estructura de la proteína; allí cada secuencia fue representada como un vector de características.

En [1] se hizo predicción de sitios de interacción proteína-proteína usando redes neuronales con funciones de base como modelo de predicción, cada secuencia fue representada como un vector de 220 atributos. También en [3] se propuso un método de predicción de sitios de interacción entre proteínas basado en redes neuronales. Por último en [13] se presentó un método para clasificación de ARN donde cada secuencia fue ajustada a un vector en un espacio euclidiano de dimensión n, se planteó un kernel [7] que involucraba la información global y local de la secuencia, para esto se consideró el número de ocurrencias de bigramas en las moléculas y se utiliza como clasificador una MSV.

2. Funciones Kernel y Kernel en Secuencias

En este capítulo se definen formalmente las funciones kernel y las transformaciones que permiten definir kernels para secuencias. También se describen los elementos de un sistema de clasificación basado en funciones kernel

2.1 Funciones Kernel

Para poder resolver un problema que no es linealmente separable se pueden utilizar las funciones kernel.

Un kernel es una función k talque para todo x, z en el espacio de entrada X satisface.

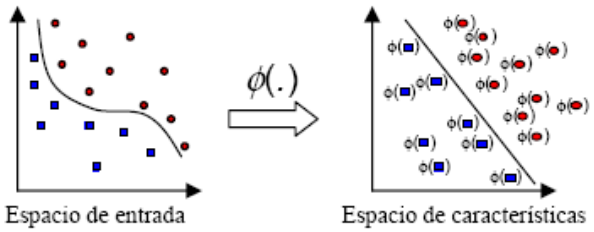
$$k(\mathrm{x}, \mathrm{z}) = \langle \phi(\mathrm{x}) \cdot \phi(\mathrm{z}) \rangle$$

donde ϕ es una transformación de X a un espacio de características F ,dotado de producto interior así

$$\phi: \mathrm{x} \mapsto \phi(\mathrm{x}) \in \mathrm{F}$$

Geométricamente, se realiza una transformación a un espacio de mayor dimensión donde se puede encontrar un separador lineal. Como se aprecia en la figura 7.

Figura 7. Representación del mapeo a un espacio separable



Es decir para esto se debe generalizar la función

$$f_{msv}(\mathrm{x}) = \sum_{\mathrm{x}_i \in VS} \alpha_i y_i \langle \mathrm{x}_i \cdot \mathrm{x} \rangle + b$$

por medio de kernel, así.

$$f_{msv}(\mathrm{x}) = \sum_{\phi(\mathrm{x}_i) \in VS} \alpha_i y_i \langle \phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}) \rangle + b$$
$$= \sum_{\phi(\mathrm{x}_i) \in VS} \alpha_i y_i k(\mathrm{x}_i, \mathrm{x}) + b$$

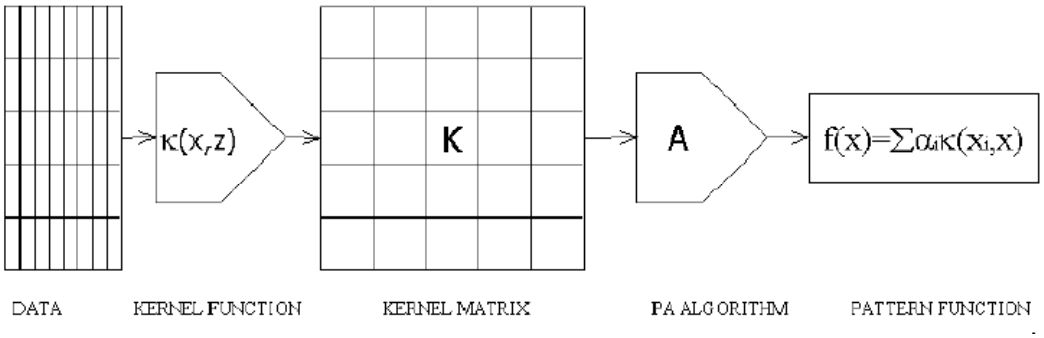
donde $k: \mathrm{X} \times \mathrm{X} \to \mathbb{R}$ es un kernel si existe $\phi: \mathrm{X} \to \mathrm{F}$ tal que

$$k(\mathrm{x}, \mathrm{z}) = \langle \phi(\mathrm{x}) \cdot \phi(\mathrm{z}) \rangle$$

para mayor ampliación de los conceptos planteados consultar [7]

2.1.1. Sistema de clasificación basado en funciones Kernel. Las etapas necesarias para desarrollar un sistema basado en funciones kernel, se resumen en la figura 8.

Figura 8. Sistema de clasificación con funciones kernel



En una etapa inicial a los datos de entrada del sistema se les aplica la función kernel seleccionada, lo cual permite calcular el producto interno en el espacio de características. Este proceso genera la matriz kernel, también llamada matriz de Gram, que sirve como elemento de entrada a un algoritmo de análisis de patrones, como por ejemplo las MSV, y generar de esta forma una función de clasificación [5].

2.2. Kernel en secuencias

El uso de kernel en cadenas de símbolos no solamente permite realizar un reconocimiento de patrones sintáctico sino además permite un análisis estadístico de patrones. Los kernel en secuencias encierran conceptos como el de alfabeto, cadena y subsecuencia, entre otros [5].

DEFINICIONES.

Un alfabeto es un conjunto finito de símbolos $\sum$.Por ejemplo, el alfabeto del español está formado por 27 símbolos. Una cadena es una secuencia finita de símbolos de $\sum$, la secuencia $s = s_1 s_2 s_3 \dots s_n$ representa una cadena de longitud n. se define además,

$$\varepsilon : cadena\ \ vacia$$
$$\sum{}^{n} : conjunto\ \ de\ \ todas\ \ las\ \ cadenas\ de\ longitud\ \ n$$
$$\sum{}^{*} = \bigcup_{n=0}^{\infty} \sum{}^{n} : conjunto\ \ de\ \ todas\ las\ Cadenas$$

Una cadena t es subcadena de s , si existen cadenas u, v tales que $s = utv$ (operación de concatenación). Si $u = \varepsilon$ se dice que t es un prefijo de s ; si $v = \varepsilon$ se dice que t es un sufijo de s.

Un k-grama es una cadena de la forma s*(i:j)* , luego u = s*(i)* denota que u es una subsecuencia de s dada por $i=(i_1,i_2,..i_u)$ y $l(i)$ se define como la longitud de la subsecuencia($i_{|u|} - i_1 + 1$)

Ejemplo:

Si $\sum = \{a, b, c...z\}$ en la secuencia s= *kernels* se tiene

s*(1:3)= ker.*

s *(1,2,4,7)= kens*

l(1,2,4,7)=7

Un concepto importante al hablar de un kernel es la similitud, podemos definir la similitud de dos secuencias de símbolos, a partir del número de cadenas contiguas de longitud p tiene en común. Se debe considerar además que para dos cadenas x, z el cálculo de la similitud, $\langle \phi(\mathrm{x}) \cdot \phi(\mathrm{z}) \rangle$, sea obtenido en forma eficiente.

Una vez definido el kernel se puede utilizar una MSV como clasificador de secuencias así:

$$f(\mathrm{x}) = \sum_{i=1}^{l} y_i \alpha_i \langle \phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}) \rangle + b$$

(± 1)

donde x_i son las secuencias de entrenamiento y_i son las etiquetas con valor ± 1

Los kernels basados en secuencias son de la forma $k: \sum \times \sum \rightarrow \mathbb{R}$ donde $\sum$ es un alfabeto. Sean $s = s_1 s_2 s_3 \ldots s_n$,y , $s' = s'_1 s'_2 s'_3 \ldots s'_n$ secuencias. Se pueden así definir varios tipos de kernels.

En general en estos tipos de kernel se busca comparar dos secuencias por medio de las subsecuencias que contienen; entre más subsecuencias tengan en común más similares son. Un elemento importante a considerar es que tales subsecuencias no necesariamente son contiguas; por ello se define el factor de decadencia λ como el grado en que la contigüidad interviene en la similitud entre las secuencias.

2.2.1 Gap- weighted subsequence Kernel. Este kernel considera el grado de separación entre secuencias. Para este kernel definimos la función de mapeo como $\boldsymbol{\phi}_{\mathbf{u}}^{p}(\mathbf{s}) = \sum_{i:\mathbf{u}=\mathbf{s}(i)} \boldsymbol{\lambda}^{l(i)}, \mathbf{u} \in \sum^{p}$

con $\lambda \in (0,1)$ factor de decadencia. El kernel se define como

$$k_p(s, s') = \langle \phi^p(s) \cdot \phi^p(s') \rangle = \sum_{u \in \sum^p} \phi^p(s) \phi^p(s')$$

Por ejemplo para las palabras cat, car bat y bar la función de transformación resultante se muestra en la tabla 3:

Tabla 3. Transformación con kernel subsequence

ϕ	ca	ct	at	ba	bt	cr	ar	br
cat	λ^2	λ^3	λ^2	0	0	0	0	0
car	λ^2	0	0	0	0	λ^3	λ^2	0
bat	0	0	λ^2	λ^2	λ^3	0	0	0
bar	0	0	0	λ^2	0	0	λ^2	λ^3

Así el kernel para k(cat,cat)= $2\lambda^4 + \lambda^6$

2.2.2. String Kernel. Para cada secuencia s se construye un vector $\boldsymbol{p}$-string(s) que contiene el número de veces que aparece cada subsecuencia en dicha secuencias. Definimos así el kernel String como

$$k_p(\mathrm{s}, \mathrm{s}') = \langle \phi^p{}_{spectrum}(\mathrm{s}) \cdot \phi^p{}_{spectrum}(\mathrm{s}') \rangle$$

Las características usadas para este kernel son el conjunto de todas la subsecuencias de aminoácidos de longitud p. Si dos proteínas tienen algunas subsecuencias iguales, su producto punto es alto. La complejidad del kernel string es $O(kp)$.

2.2.3. Full string Kernel. Este kernel es una extensión del string kernel. No solo se consideran subsecuencias de longitud fija $\boldsymbol{p}$, sino además todas las posibles subsecuencias de longitud uno hasta las de longitud $\boldsymbol{p}$. El full string kernel se calcula de igual forma que el string kernel.

2.2.4. Kernel identidad. Este kernel cuenta el número de residuos en que coinciden las secuencias $\mathbf{s}$ y $\mathbf{s}'$ [18]

$$k(\mathrm{s}, \mathrm{s}') = \sum_{k=1}^{n} e(s_k, s'_k)$$

donde e es una función definida como $e(s_k, s'_k) = \begin{cases} 1\,, si\ s_k = s'_k \\ 0\,, si\ s_k \neq s'_k \end{cases}$

2.2.5. Substitution Kernel. Considere la matriz de sustitución $\mathbf{M}$, (BLOSUM62 o HENS920102) [33], y el menor y mayor valor de la matriz denotados min, máx. respectivamente.

Definimos

$$\mathbf{M}'(i,j) = \frac{\mathbf{M}(i,j) - min}{máx - min}$$
$$\mathbf{M}'(i,i) = \mathbf{M}'(i,i) - \lambda$$

Donde λ es el menor valor propio de $\mathbf{M}'$. Esta operación se realiza para obtener una matriz semidefinida positiva $\mathbf{M}'$ [12].

Así, el kernel resultante es $k(s,s') = \sum_{k=1}^{n} \mathbf{M}'(s,s')$

3. Experimentación con las funciones kernel

En la primera sección de este capítulo se presenta la descripción del conjunto de datos experimentales y el preprocesamiento realizado a los mismos. En la última sección del capítulo se muestran los resultados del desempeño obtenidos al realizar un análisis exploratorio de los datos. Con esta prueba se pretende evaluar la adecuada implementación de las funciones kernel para secuencias y la comprensión experimental de la variación de los parámetros en dichas funciones. Las funciones kernel aplicadas en esta prueba fueron kernel string, full string y kernel identidad

3.1. Descripción del conjunto de datos

El conjunto de datos utilizados en este trabajo se denomina *Catalytic Site Atlas* (CSA)[1]. Esta es una base de datos que contiene anotaciones de sitios activos y residuos catalíticos de enzimas en estructura 3D, tomadas de la base de datos PDB (Protein Data Bank). La última versión 2.2.12, actualizada en enero del 2010, contiene 968 enzimas.

Este conjunto contiene enzimas con estructura conocida y sitios activos bien definidos. Las enzimas fueron tomadas de la base de datos de estructuras de enzimas[2].

El CSA consiste de dos tipos de anotaciones, un conjunto de información de la literatura primaria y un conjunto adicional de homólogas, que contiene anotaciones inferidas por PSI-BLAST. La versión 1.0 de CSA [48] contiene dos tipos de

[1]http://www.ebi.ac.uk/thornton-srv/databases/CSA/
[2]http://www.biochem.ucl.ac.uk/bsm/enzymes/index.html

entradas, un conjunto original de 177 enzimas y un conjunto de 2608 entradas homólogas que cubren menos del 30% de todas las enzimas en PDB. Con estos dos tipos de entradas se tiene un total de 17917 residuos

3.2. Preprocesamiento de los datos

La primera etapa del preprocesamiento hace referencia a la integración de los datos ya que se tiene un primer archivo de texto proporcionado por CSA**,** en el cual se indica el identificador de la enzima, el residuo y la posición del residuo catalítico de la enzima y la fuente de información, por ejemplo, para la enzima 1a0i se tiene

1a0i,0,Lys,None,34,s,LIT.

Un segundo archivo con las secuencias en formato FASTA en donde la primera línea ó cabecera proporciona el identificador de la enzima las líneas siguientes, cada una con máximo 80 caracteres, corresponden a la secuencia de la enzima identificada, como se aprecia para 1a0i.

>1A0I:_|PDBID|CHAIN|SEQUENCE

VNIKTNPFKAVSFVESAIKKALDNAGYLIAEIKYDGVRGNICVDNTANS
YWLSRVSKTIPALEHLNGFDVRWKRLLNDDRCFYKDGFMLDGELMVKG
VDFNTGSGLLRTKWTDTKNQEFHEELFVEPIRKKDKVPFKLHTGHLHIKL
YAILPLHIVESGEDCDVMTLLMQEHVKNMLPLLQEYFPEIEWQAAESYEV
YDMVELQQLYEQKRAEGHEGLIVKDPMCIYKRGKKSGWWKMKPENEAD
GIIQGLVWGTKGLANEGKVIGFEVLLESGRLVNATNISRALMDEFTETVKE
ATLSQWGFFSPYGIGDNDACTINPYDGWACQISYMEETPDGSLRHPSFVMFR

Con estos dos archivos se realizó una limpieza de los datos debido a que se tenían secuencias con símbolos que no corresponde a los permitidos para

secuencias de aminoácidos ó posiciones que no están dentro del rango de la longitud de la secuencia

Posteriormente, considerando que los residuos catalíticos que determinan un sitio activo no se encuentran cercanos en secuencia y el hecho que la mayoría de las enzimas tiene de uno a cinco residuos que intervienen en la catálisis [19]**,** se decidió no tomar la totalidad de la secuencia debido a que se genera mayor dificultad en el proceso de predicción de los residuos catalíticos, pues un gran número de los residuos en una enzima no intervienen en el proceso de catálisis.

De esta forma para generar el conjunto de datos de entrada de la clase positiva (+1) para la MSV se tomaron ventanas ó subsecuencias de 11 residuos contiguos, de tal forma que el residuo catalítico tuviera en cada lado cinco aminoácidos, como se planteó en [12] [20].Por cada residuo catalítico presente en una enzima se generó una ventana de esta longitud. De igual manera se generó el conjunto de datos de entrada de la clase negativa (-1), pero con la salvedad de que los residuos que conforman cada ventana no contuvieran residuos catalíticos.

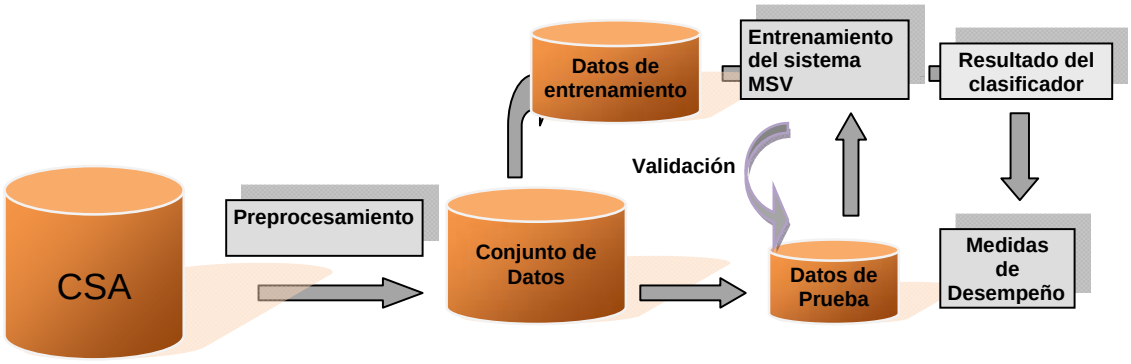
Con el fin de obtener un conjunto de datos balanceado por cada entrada de clase positiva se generó una entrada de clase negativa en la misma enzima.

3.3. Análisis exploratorio de los datos

El objetivo de realizar este análisis exploratorio de datos fue el de obtener una medida de desempeño preliminar del modelo de clasificación de residuos catalíticos y, además, determinar de qué manera los parámetros definidos en las secciones 1.3.4 y 2.2: factor de decadencia (λ), longitud de la subsecuencia (n) y la complejidad de la MSV (C), afectan el desempeño del clasificador.

Para generar el conjunto de datos de entrada al sistema de clasificación se consideraron 430 secuencias de enzimas de la base de datos CSA. Una vez realizado el preprocesamiento descrito en la sección 3.2 se obtuvo un total de 1000 secuencias**,** 500 de la clase positiva(+1) y 500 de la clase negativa(-1); posteriormente se dividió el conjunto total de datos en dos grupos uno de entrenamiento con 700 secuencias y otro de prueba con las restantes 300 secuencias. El sistema de clasificación se entrenó utilizando la técnica de validación cruzada con 10 grupos. La metodología utilizada para este análisis se presenta gráficamente en la figura 9.

Figura 99. Metodología para el análisis exploratorio de los datos



3.3.1. Resultados experimentales. Para cada una de estas pruebas se entrenaron varias MSV usando el software R versión 2.8.1 y la librería kernlab. El equipo de cómputo utilizado tenía las siguientes características HP pavilion 6353 con procesador AMD Athlon x2 de 2GHz, 3GB de memoria y cache de 1000k.

- Resultados con el kernel string

Una primera prueba se realizó fijando el factor de decadencia λ en 0.5, considerando que este parámetro debe estar comprendido entre 0 y 1. Un valor de λ cercano a uno, indica que no importa que tan lejos estén las subsecuencias, y un valor cercano a cero indica que tiene en cuenta que tan lejos están.

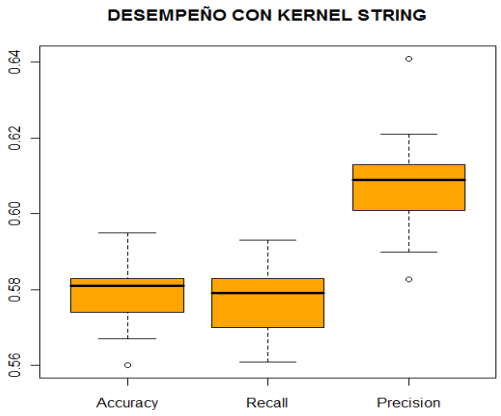
Una vez fijado el factor de decadencia se varía la longitud de las subsecuencias (n) y la complejidad del modelo (C) para cada una de las funciones kernel string, full string, sequence y kernel identidad. En la tabla 4, se indican los errores de validación para cada uno de los diferentes valores del kernel string.

Tabla 4. Error de validación para el kernel string

kernel string	***C*=2**	***C*=4**	***C*=8**	***C* =10**	***C* =16**
n=2	0.436	0.446	0.47	0.401	0.4066
n=3	0.4667	0.46	0.475	0.46	0.4616

Como se aprecia en la tabla4, el menor error de validación se tienen para subsecuencias de longitud dos (n=2), y C con un valor de 10. Al aplicar el modelo generado con el kernel string a los datos de prueba se tiene que la mejor exactitud es de 59.5% y se obtuvo cuando C tomó el valor 16. Las medidas estadísticas del desempeño del modelo obtenido al realizar 15 ejecuciones con validación cruzada se muestran en la figura 10.

Figura 10. Estadísticas de desempeño con kernel string



- Kernel full string.

La tabla 5 muestra los errores de validación para los modelos generados fijando el factor de decadencia para el kernel full string y variando los otros dos parámetros.

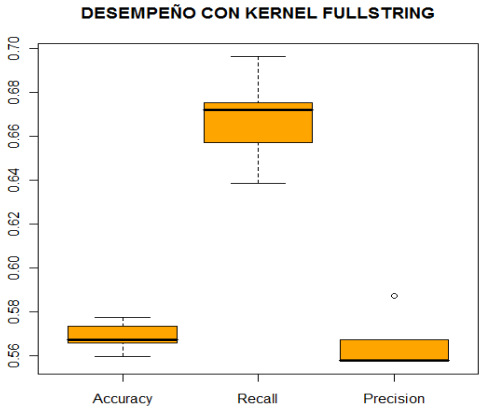
Tabla 5. Error de validación para kernel full string

kernel full string	***C* =2**	***C* =3**	***C* =4**
n=2	0.4066	0.4066	0.4066
n=3	0.4133	0.4140	0.4150

El menor error de validación se tiene para los parámetros n igual a 2 y C igual a 2. La exactitud del modelo generado es de 57.5%; cabe notar que este valor también se tiene para un modelo con kernel full string con n igual 3 y C igual a 2. Con esta experimentación cabe destacar que el desempeño obtenido es más bajo que el

proporcionado por el kernel string, y además que el tiempo de procesamiento resulta ser mayor. Las estadísticas de desempeño al realizar 10 ejecuciones se muestran en la figura 11.

Figura 10. Medidas de desempeño para el kernel full string



- Kernel sequence

En la tabla 6 se presentan los errores de validación para las subsecuenciasde longitud dos y tres. Se tiene que el menor error de validación es 0.43 y se obtuvo con un valor para C igual 4 y un valor para n igual a 3, como se aprecia a continuación

Tabla 4. Error de validación para el kernel sequence

kernel sequence	***C* =2**	***C* =3**	***C* =4**
n=2	0.435	0.435	0.4365
n=3	0.440	0.441	0.4300

Un modelo de clasificación con los parámetros que generan el menor error de validación produce una medida de exactitud de 58.5%, precisión de 0.651 y sensibilidad de 0.425.

- Kernel identidad

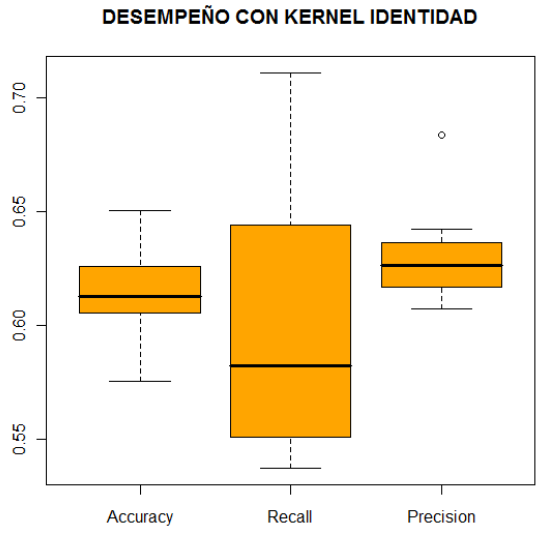
El último kernel aplicado a este análisis exploratorio fue el kernel identidad, en el cual no se cuenta con el parámetro referente a la longitud de la subsecuencia (n). En la tabla 7 se muestra el error de entrenamiento y de validación para cada modelo con diferentes valores de complejidad

Tabla 7. Error de validación para el kernel identidad

kernel identidad	C **=2**	C **=3**	C **=4**	C **=5**
Error de entrenamiento	0.1651	0.1013	0.1223	0.1031
Error de validación	0.4014	0.436	0.437	0.4389

El error de validación fue menor para C igual 2; sin embargo, el mayor valor de exactitud fue de 65% y se obtuvo con el parámetro C igual a 3, así el desempeño se resume en la figura 12.

Figura 11. Medidas de desempeño para el kernel identidad



De este análisis exploratorio se observó que para este conjunto de 1000 secuencias el modelo con mejor desempeño se logró con el kernel identidad;

siendo la exactitud de 65%, que no es un valor muy alto pero si aceptable comparado con resultados obtenidos en trabajos de enfoques similares [12].

Se concluye con esta experimentación que para los kernels string, sequence y full string, el factor de decadencia no afecta significativamente el modelo. Por último la longitud de las subsecuencias para los kernels string y full string debe estar comprendida entre dos y cinco, ya que para valores superiores a cinco el modelo los clasifica en una misma clase.

Ahora el objetivo es aplicar técnicas de aprendizaje de máquina que permitan mejorar el desempeño del modelo obtenido en la etapa de análisis preliminar.

4. Métodos basados en Kernel para la clasificación de residuos catalíticos en sitios activos de enzimas

En este capítulo se describe una metodología para la clasificación de residuos catalíticos que permite aumentar el desempeño del clasificador obtenido en el capítulo anterior. Dicha metodología además de considerar un modelo de clasificación basado en una MSV, aplica una técnica de agrupamiento, definida en la sección 1.3.1, con lo cual se busca formar grupos según la naturaleza de las enzimas.

En las primeras dos secciones se describen en forma detallada cada una de las etapas de la metodología planteada. En la tercera sección se presenta la técnica utilizada para determinar la complejidad del modelo que minimiza el error de validación. Por último se tiene la sección con los detalles de la implementación.

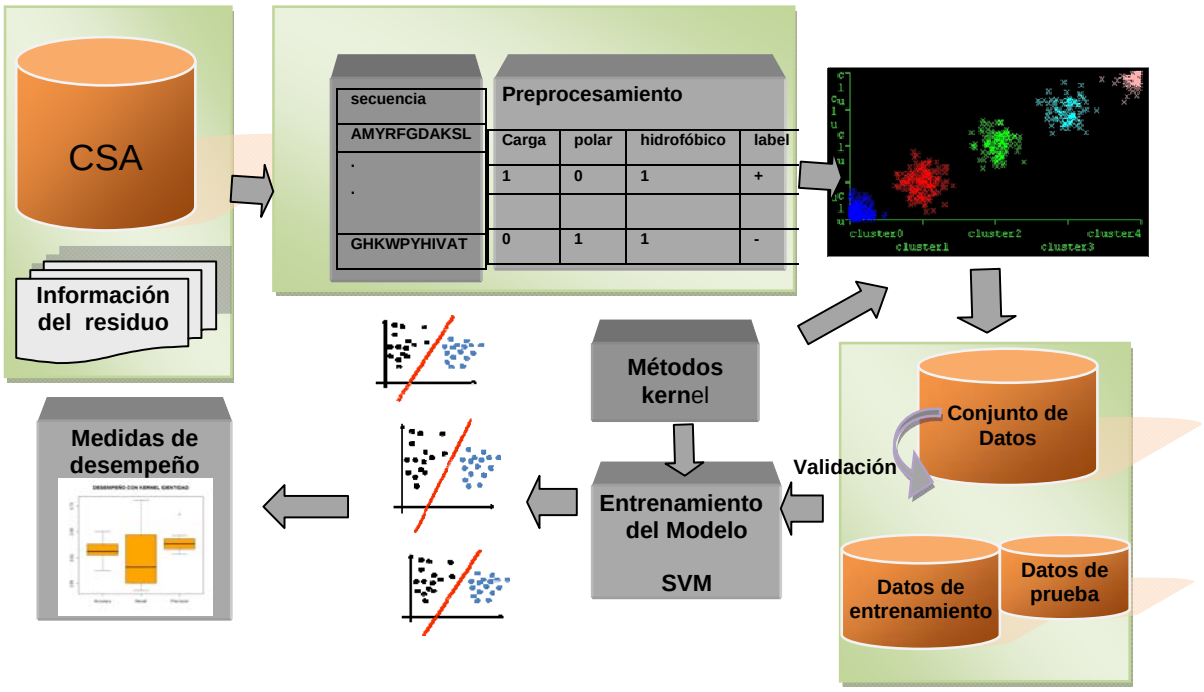
Dependiendo de la reacción que catalice una enzima puede ser clasificada en una de seis categorías principales [37], descritas en la sección 1.2.2. Debido a este hecho, es posible trazar una metodología que logre involucrar esta información, por medio de aprendizaje no supervisado, y que permita mejorar el desempeño de un clasificador de residuos catalíticos basado en funciones kernel, como se plantea en [21].

Esta metodología se puede resumir a partir de la figura 13 donde se muestra la etapa de preprocesamiento de los datos, en la cual se busca representar e integrar los datos obtenidos de la base de datos CSA y las anotaciones.

Posteriormente se realiza la etapa de agrupación, cuyo objetivo es determinar relaciones existentes entre las secuencias que contienen residuos catalíticos. Cabe destacar que los métodos kernel cobran gran importancia en esta etapa, debido a que los datos con que se cuenta son de diversa naturaleza.

Por último, se tiene la etapa de entrenamiento y validación del modelo de clasificación. En esta etapa se entrena una MSV, basados en la técnica de validación cruzada, por cada grupo obtenido en el agrupamiento de los datos. Cabe notar que las funciones kernel utilizadas en la MSV no consideran la información fisicoquímica de los residuos catalíticos, ya que se busca que el modelo clasifique un residuo catalítico solamente a partir de la secuencia que lo contiene. Además se determinan los parámetros del modelo de clasificación que permitan obtener un mejor desempeño del mismo.

Figura 12. Metodología para la clasificación de residuos catalíticos.



Cada etapa de la metodología utilizada es descrita en detalle a continuación.

4.1. Preprocesamiento

El preprocesamiento de los datos se puede presentar en dos etapas. Una primera etapa hace referencia al proceso descrito en la sección 3.3, en el cual no se considera toda la secuencia como entrada al sistema de clasificación, si no que se generan ventanas o subsecuencias de 11 residuos contiguos.

En una segunda etapa, además de la información en secuencia de *Catalytic Site Atlas* (CSA) se hace uso de información biológica de cada residuo, referente a las propiedades de su cadena lateral y las propiedades fisicoquímicas de los aminoácidos [48]. En particular se tomaron características como:

- Tipo de residuo: Los residuos se pueden dividir en tres grupos principales [8]. Los de tipo carga, en donde se tienen los aminoácidos (H,R,K,E,D,); los de tipo polar conformado por (Q,T,S,N,C,Y,W) y, por último, los de tipo hidrofóbico, que incluye los aminoácidos restantes (G,F,L,M,A,I,P,V)

- Masa: La masa atómica puede ser considerada como la masa total de protones y neutrones en un solo átomo. Frecuentemente expresada en unidades de masa atómica unificada. Equivale a 1/12 parte de la masa de un átomo de carbono-12 [29].

- Superficie: el área de la superficie es importante ya que la interacción de los residuos con otras moléculas sucede en la superficie. se consideró la superficie topográfica [2] [29].

- Volumen: el volumen de cada aminoácido se obtuvo de [30]

En una segunda prueba de agrupación, estas características fisicoquímicas de los residuos se tuvieron en cuenta para generar los grupos; con el fin de medir si tales características mejoran la medida de similitud en los grupos. Sin embargo, para el entrenamiento del clasificador no fueron tomadas en cuenta dichas propiedades físico-químicas, debido a que se busca clasificar ó determinar si una enzima contiene un residuo catalítico solamente a partir de su secuencia.

Cabe tener presente que la fuente de datos sobre el cual se realizó el preprocesamiento es la misma que se utilizó en análisis experimental del capítulo anterior; sin embargo, el conjunto de secuencias con residuos catalíticos se incrementó a 1620, los cuales fueron obtenidos de 523 enzimas.

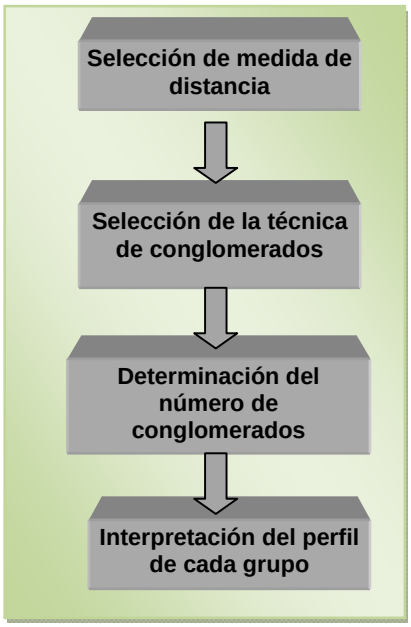
4.2. Análisis de agrupamiento

La idea básica del agrupamiento es particionar un conjunto de datos en subgrupos, basados en una medida de similitud, de tal manera que los elementos de un subgrupo tengan mayor similitud entre sí que con los elementos de otro subgrupo [22].

Con el agrupamiento realizado a las 1620 secuencias, se pretende buscar interrelaciones existentes entre los diversos aminoácidos que permitan entender la distribución o estructura de los residuos catalíticos en enzimas.

El procedimiento realizado para el proceso de agrupamiento se presenta en la figura 14, y se describe a continuación.

Figura 13. Procedimiento de agrupación



4.2.1. Selección de la medida de distancia.

En el espacio de características F sobre el cual se transforman los datos de entrada $D = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_l\}$,$D \subseteq \mathrm{X}$. Se define la norma de $\phi(\mathrm{x})$ como

$$\|\phi(\mathrm{x})\| = \sqrt{\langle \phi(\mathrm{x}) \cdot \phi(\mathrm{x}) \rangle} = \sqrt{K(\mathrm{x}, \mathrm{x})}$$

Se define la distancia al cuadrado entre dos vectores $\phi(\mathrm{x}_i)$,y , $\phi(\mathrm{x}_j)$ en el espacio de características F como.

$$\begin{aligned} \left\|\phi(\mathrm{x}_i) - \phi(\mathrm{x}_j)\right\|^2 &= \langle \phi(\mathrm{x}_i) - \phi(\mathrm{x}_j) \cdot \phi(\mathrm{x}_i) - \phi(\mathrm{x}_j) \rangle \\ &= \langle \phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}_i) \rangle - 2\langle \phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}_j) \rangle + \langle \phi(\mathrm{x}_j) \cdot \phi(\mathrm{x}_j) \rangle \\ &= K(\mathrm{x}_i, \mathrm{x}_i) - 2K(\mathrm{x}_i, \mathrm{x}_j) + K(\mathrm{x}_j, \mathrm{x}_j) \end{aligned}$$

donde ϕ transforma cada dato de entradas x en un elemento $\phi(x)$ en el espacio F

$$\phi: \mathrm{x} \mapsto \phi(\mathrm{x}) \in \mathrm{F}$$

La conformación de los diversos grupos se basa en una medida de similitud, en especial, la medida de similitud que se utilizó es la distancia entre la

transformación obtenida de cada una de las secuencias en el espacio de características, y que se calcula a partir de la función de kernel establecida. Se puede obtener así una matriz de distancias entre los vectores que conforman el espacio de características, de igual manera, como se obtiene la matriz de kernel definida en la sección 2.1.1.

4.2.2. Selección de técnica de conglomerados. Una de las etapas de mayor importancia en el análisis de conglomerados, es la determinación del procedimiento mediante el cual se agrupen los datos más similares dentro de un conglomerado, que permita especificar la pertenencia de cada secuencia a un grupo. Para entender el método *k*-medias [22] aplicado a datos que no se representan en forma vectorial, se deben tener presente las siguientes definiciones.

El centro de masa en el espacio de características F de un conjunto de datos de entrada $D = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_l\}$, se define como $\phi_D = \frac{1}{l}\sum_{i=1}^{l}\phi(\mathrm{x}_i)$

La distancia de la transformación de un dato x al centro de masa está dada por

$$\|\phi(\mathrm{x}) - \phi_D\| = \sqrt{\langle\phi(\mathrm{x})\cdot\phi(\mathrm{x})\rangle + \langle\phi_D\cdot\phi_D\rangle - 2\langle\phi(\mathrm{x})\cdot\phi_D\rangle}$$
$$= \sqrt{K(\mathrm{x},\mathrm{x}) + \frac{1}{l^2}\sum_{i,j=1}^{l}K(\mathrm{x_i},\mathrm{x}_j) - \frac{2}{l}\sum_{i=1}^{l}K(\mathrm{x},\mathrm{x}_j)}$$

Como el método *k*-medias requiere un número inicial de grupos, se puede recurrir primero a aplicar un algoritmo jerárquico como AGNES, que permita dar una idea del número adecuado de conglomerados, según el coeficiente de aglomeración.

En este trabajo se emplean ambas técnicas; se aplica primero un algoritmo jerárquico AGNES y luego un algoritmo particional como lo es *k*-medias.

4.2.3. Determinación del número de grupos. Una de las dificultades del análisis de conglomerados es determinar el número adecuado de grupos a considerar; ya que en los algoritmos jerárquicos este número depende de donde se corte el árbol de jerarquías; y en los algoritmos no jerárquicos se debe especificar la cantidad de grupos o particiones del conjunto de datos a priori.

Para determinar el número de grupos que permita conocer la mejor distribución de los datos en conglomerados; se debe tener una medida de conglomeración que permita evaluar el desempeño de los algoritmos empleados.

Algunas medidas de aglomeración son:
Coeficiente aglomerativo: asociada a los métodos jerárquicos. Así, para cada dato x_i del conjunto de entrada $D = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l\}$ se define $d(\mathrm{x}_i)$ como la distancia al primer conglomerado con que se une, dividida por la distancia de los últimos conglomerados en unirse. El coeficiente aglomerativo es

$$\mathrm{CA} = 1 - \frac{\sum_{i=1}^{l} d(\mathrm{x}_i)}{l}$$

Cuadrado medio dentro de los conglomerados (CMD): Es el promedio de la suma de cuadrados de las distancias de cada observación hasta el centroide del conglomerado al que pertenecen.

$$\mathrm{CMD} = \frac{1}{d}\sum_j \left(\sum_i \left\| \mathrm{x}_{ij} - \bar{\mathrm{x}}_j \right\|^2\right)$$

donde x_{ij} representa el dato x_i del conglomerado j.

$d = \sum n_j$

(n_j numero de datos en conglomerado j)

4.3. Resultados análisis de clúster en secuencias

Para estas pruebas el conjunto total de datos se dividió en un conjunto de entrenamiento y validación, conformado por 1219 secuencias con residuos catalíticos, sobre el cual se realizaron las pruebas de análisis de conglomerados; y un conjunto de validación de 300 secuencias. Esta partición del conjunto de datos se realizó debido a que la metodología de validación cruzada para el entrenamiento de la MSV así lo requiere.

El procedimiento de agrupación se realizó para dos escenarios diferentes. Un primer escenario en el cual los datos de entrada corresponden a la secuencia de cada enzima, la cual contiene un residuo catalítico. El segundo escenario, además de la secuencia, cuenta con información fisicoquímica del residuo catalítico que contiene.

Los resultados obtenidos del procedimiento de agrupación en cada uno de estos escenarios se muestran a continuación.

4.3.1. Agrupamiento basado en secuencia. Como se estableció en la sección 4.2.1, cada función kernel determina una matriz de distancias en el espacio de características. Teniendo presente este hecho, se consideró una matriz inducida por el kernel string y una segunda matriz obtenida al aplicar el kernel identidad, porque estos tipos de kernels lograron mejor desempeño en la sección de resultados experimentales.

Al aplicar el algoritmo de conglomerado AGNES a cada matriz de distancias considerada, se obtuvieron los siguientes arboles jerárquicos o dendogramas, los cuales darán una idea del número de grupos en los cuales se pueden particionar el conjunto de residuos catalíticos de interés.

La figura 15 muestra el dendograma al utilizar el kernel string con el parámetro n=3. Como se aprecia, en un conjunto muy grande de datos el realizar un análisis más detallado sobre la conformación de cada grupo a partir del dendograma es un poco complicado; sin embargo, al revisar el diagrama de barras, figura 16,se puede determinar un número de grupos en el que se puede particionar el conjunto.

Figura 14. Dendograma con kernel string solo secuencia

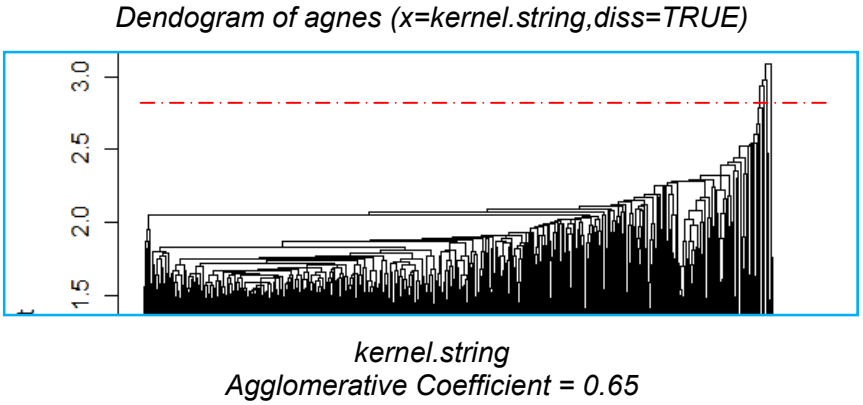
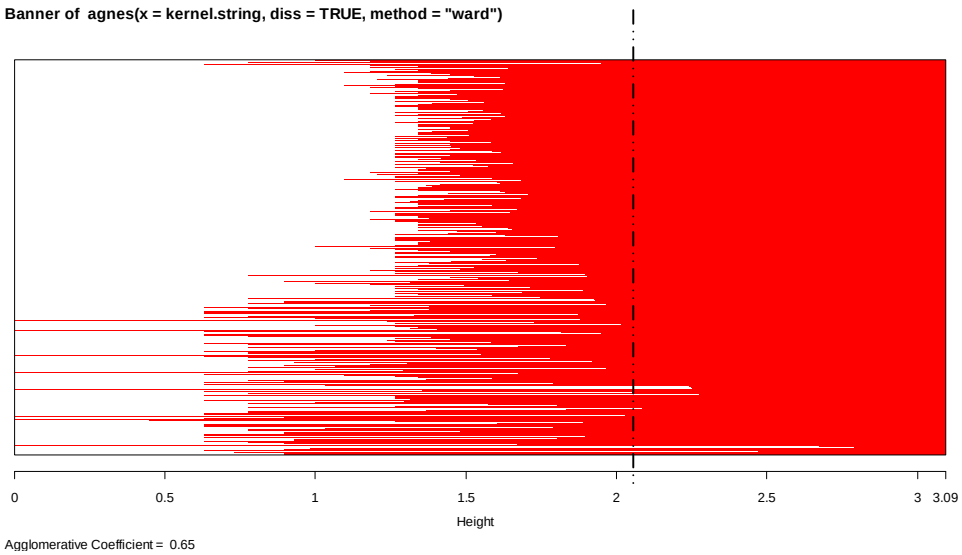


Figura16. Diagrama de barras con kernel string



Al trazar una vertical en el valor 2.0, se corta el diagrama de barras en 4 puntos, lo cual indica que se pueden realizar una partición en igual número de grupos.

La figura 17 muestra el dendograma al utilizar el kernel identidad. Allí se aprecia que al realizar el corte a una altura aproximada de 10, el árbol jerárquico es

cortado en 6 puntos, lo cual se puede interpretar como que en el conjunto de las 1320 secuencias de enzimas con residuos catalíticos utilizado se identifican 6 conglomerados. Cabe destacar que el coeficiente de aglomeración de este dendograma es mayor que el del obtenido con el kernel string.

Figura 17. Dendograma con kernel identidad basado en secuencia

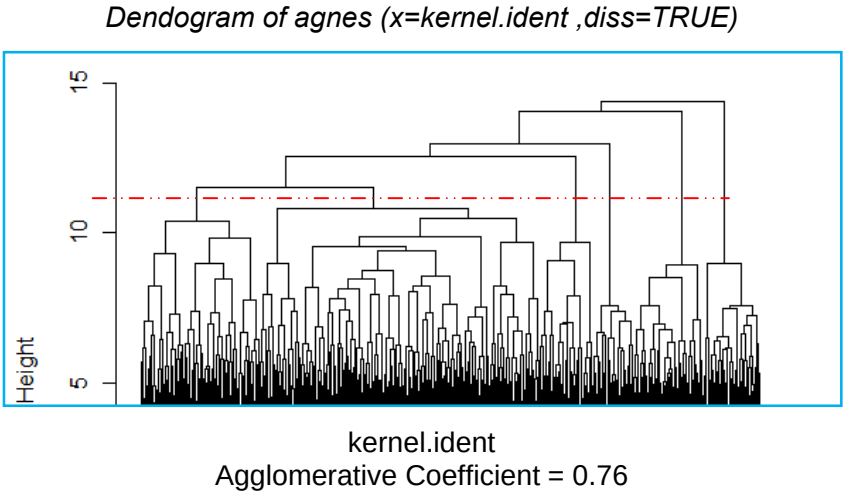
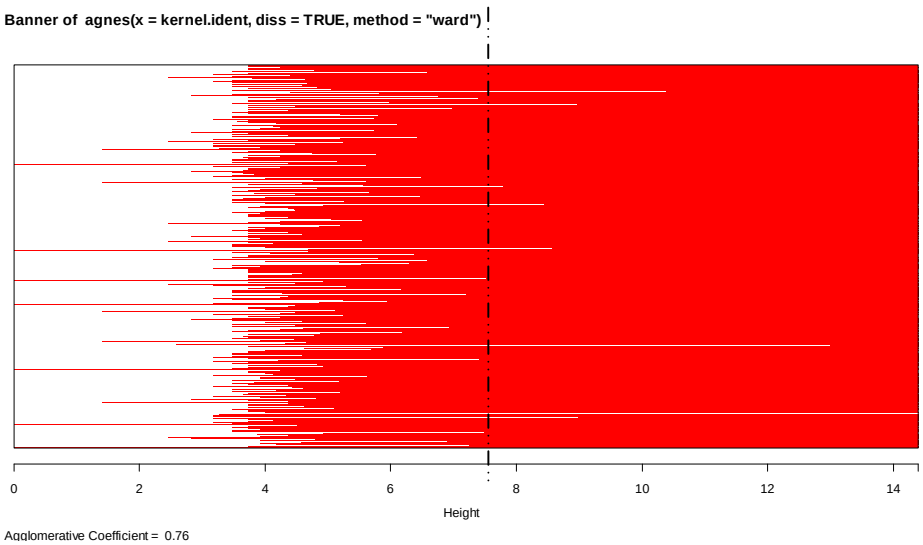


Figura 18. Diagrama de barras con kernel identidad



La diferencia que se presenta en los resultados obtenidos, al aplicar un algoritmo jerárquico con diferentes funciones kernel, se puede explicar por la poca similitud entre las matrices kernel resultantes. Para verificarlo se puede definir una medida

de similitud entre dos matrices de kernel (matrices de Gram), considerando el producto interno de Frobenius [28] entre dos matrices de igual dimensión, el cual se define enseguida.

Definición: sean $\mathbf{A}$ y $\mathbf{B}$ matrices de igual dimensión $\mathrm{l} \times \mathrm{l}$. El producto interno de Frobenius $\langle \mathbf{A}, \mathbf{B} \rangle_F$ está dado por $\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_{i,j=1}^{\mathrm{l}} \mathbf{A}_{ij} \mathbf{B}_{ij} = tr(\mathbf{A}'\mathbf{B})$.

Aplicando esta definición a las matrices kernels, se tiene una medida de similitud entre ellas como se define a continuación.

Definición: Sean $\mathbf{K_1}$,y, $\mathbf{K_2}$ dos matrices kernel. $D = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l\}$ un conjunto de entrada. El alineamiento $A(\mathbf{K_1}, \mathbf{K_2})$ [25] [27] entre las matrices kernel está dado por D

.

$$A(\mathbf{K_1}, \mathbf{K_2}) = \frac{\langle \mathbf{K_1}, \mathbf{K_2} \rangle_F}{\sqrt{\langle \mathbf{K_1}, \mathbf{K_1} \rangle_F \langle \mathbf{K_2}, \mathbf{K_2} \rangle_F}}$$

El valor de alineamiento satisface que $-1 \leq A(\mathbf{K_1}, \mathbf{K_2}) \leq 1$; un valor de 1 indica que los kernels están alineados, es decir, que las matrices kernel son la misma. Un valor de 0 indica que no están correlacionadas.

El alineamiento entre las matrices $\mathbf{K_1}$ = kernel string y $\mathbf{K_2}$ = kernel identidad arrojó un valor $A(\mathbf{K_1}, \mathbf{K_2}) = 0.29$, lo cual indica que las matrices $\mathbf{K_1}$ y $\mathbf{K_2}$ no son la misma, es decir existe diferencia cuando se utiliza un kernel string y un kernel identidad en el algoritmo jerárquico AGNES.

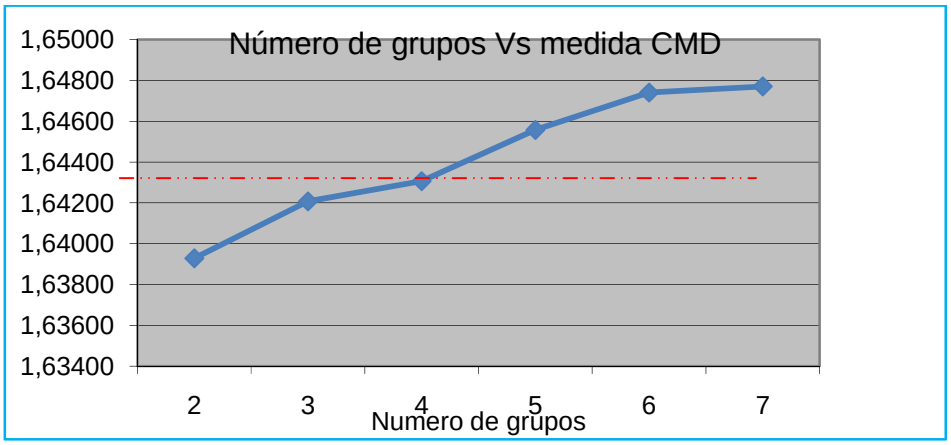
El siguiente paso para determinar el número adecuado de grupos en los cuales se debe particionar el conjunto de residuos catalíticos, de tal manera que sea posible establecer las relaciones existentes entre los aminoácidos, es aplicar el algoritmo

k-medias, tomando como parámetro para dicho algoritmo el número de conglomerados obtenido al aplicar AGNES.

El algoritmo k-medias realiza una partición aleatoria inicial de las secuencias, según el parámetro (n), que corresponde al número de grupos; por ello se realizaron15 iteraciones por cada valor de n, con el fin de seleccionar los conglomerados que minimicen el error cuadrático medio, para una partición establecida como adecuada.

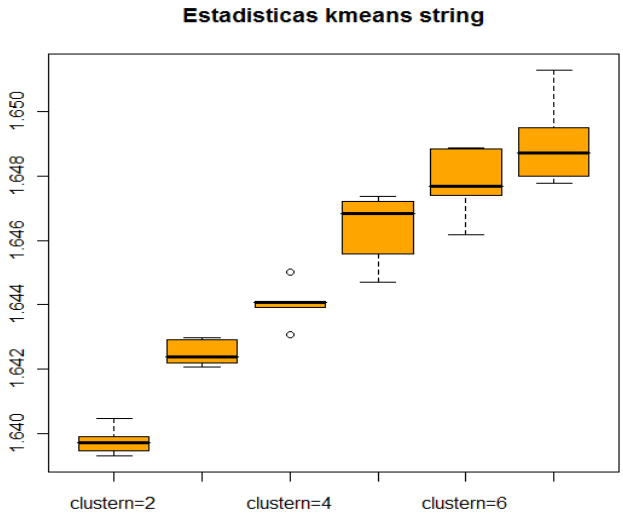
La figura 19 muestra el error cuadrático medio obtenido al aplicar k-medias con el kernel string, para un número de grupos que varía entre dos y siete.

Figura 19. k-medias con kernel string para secuencia



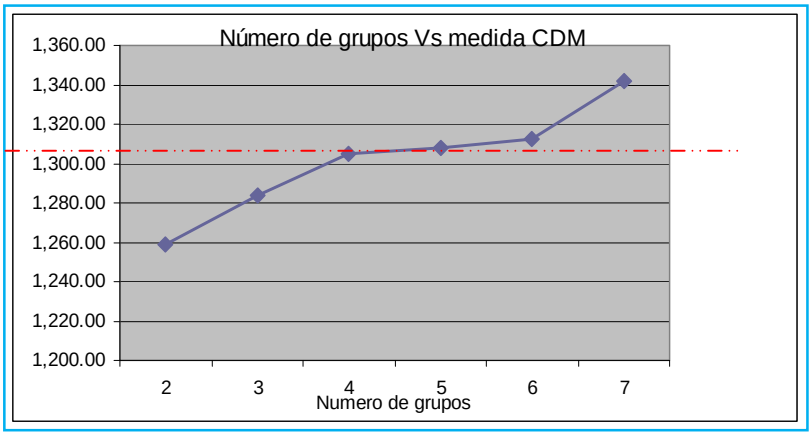
En la gráfica anterior se aprecia que para una partición entre tres y cuatro grupos, el error cuadrático tiene menor variación. Lo cual indica que este es un número adecuado de grupos en los cuales se pueden particionar los datos.
La figura 20 muestra la media y la varianza del error cuadrático medio obtenida al realizar las 15 iteraciones para cada uno de los grupos; en ella se aprecia que la menor varianza se dio para una partición en cuatro grupos.

Figura 15. Estadísticas error cuadrático para *k*-medias-string.



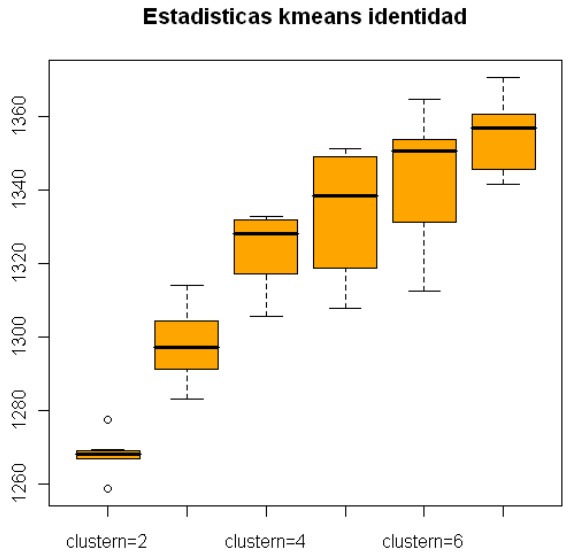
La figura 21 muestra el error cuadrático medio al aplicar el algoritmo k-medias, pero en esta ocasión se utilizó el kernel identidad. Se puede apreciar en la gráfica que un número adecuado de grupos en los cuales se pueden particionar los datos estaría comprendido entre 5 o 6 grupos, lo cual está acorde con los resultados obtenidos al analizar la figura 18.

Figura 16. k-medias con kernel identidad para solo secuencia



La figura 22 resume los valores de la media y varianza al realizar 15 iteraciones del algoritmo k-medias para cada uno de los grupos. Para particiones de los datos en cinco grupos, la varianza del error cuadrático fue la mayor; la menor varianza se dio para particiones en dos grupos, pero esto no indica que esta sea la mejor partición, ya que se busca la menor variación entre los promedios de las iteraciones de cada grupo.

Figura 172. Estadísticas error cuadrático para k-medias-identidad



4.3.2. Agrupamiento con secuencia y propiedades fisicoquímicas.

Para esta etapa del análisis de los grupos, además de las secuencias que contienen el residuo catalítico, se tiene como datos de entrada la información físico- química de cada residuo catalítico, descrita en la sección 4.1. Esta experimentación se realizó para determinar si al incluir la información biológica se obtiene una mejor conformación de los grupos.

Es decir se tiene dos fuentes de datos de entrada de diferente naturaleza; una fuente que proporciona la secuencia, CSA, y otra fuente [31], que permite determinar un vector de características de dimensión seis que contiene información respecto a las propiedades fisicoquímicas de cada residuo.

Cada tipo de datos de entrada define una función kernel diferente; para los datos proporcionados por CSA se considera un kernel basado en secuencia, kernel string (denotado por k_1). Para la información biológica ó fisicoquímica se considera un kernel gaussiano (notado k_2), definido como:

Para todo x, z en el espacio de entrada $\mathrm{X} \subseteq \mathbb{R}^{\mathrm{n}}$. El kernel gaussiano [34] está dado por

$$k_2(\mathrm{x}, \mathrm{z}) = \exp\left(\frac{-\|\mathrm{x} - \mathrm{z}\|^2}{2\sigma^2}\right)$$

donde $\sigma \in \mathbb{R}^+$.

Las funciones kernel satisfacen algunas propiedades, conocidas como propiedades de clausura, las cuales permiten definir kernels más complejos [32] o combinar funciones kernel de diferentes fuentes de datos. Algunas de estas propiedades son [35].

Propiedades de clausura: Si k_1 y k_2 son funciones kernel, y $x, z \in X$, X espacio de entrada, las siguientes funciones son kernels.

i) $k_{suma}(x,z)=k_1(x,z)+k_2(x,z)$

ii) $k(x,z)= \alpha k_1(x,z),\ \alpha \in \mathbb{R}^+$

iii) $k(x,z)=k_1(x,z)k_2(x,z)$

Para esta etapa se considera la función kernel dada por la suma de los kernels de las respectivas entradas; esta operación se puede ver básicamente como la suma de los productos internos [33]. Para el kernel gaussiano se tomó como parámetro $\sigma = 0.5$

Una vez conformado este nuevo kernel, se hace necesario normalizarlo para evitar el sesgo producido por la diferente naturaleza de los datos de entrada. Y además, para evitar que las características con rangos numéricos grandes dominen las otras características.

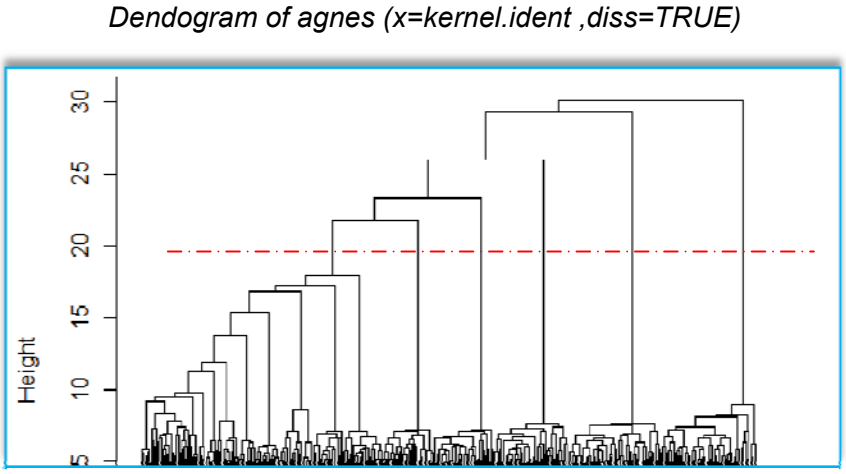
La normalización del kernel se obtiene al definir un nuevo mapeo en el espacio de características, el cual se puede describir como una normalización de los vectores en el espacio de características. Así.

En el espacio de características F, sobre el cual se transforman los datos de entrada X, se puede definir una nueva transformación $\hat{\phi} = \frac{\phi(x)}{\|\phi(x)\|}$, y de esta manera definir un nuevo kernel normalizado como:

$$\hat{k}(x,z) = \left\langle \hat{\phi}(x), \hat{\phi}(z) \right\rangle = \left\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(z)}{\|\phi(z)\|} \right\rangle = \frac{k(x,z)}{\sqrt{k(x,x)k(z,z)}}$$

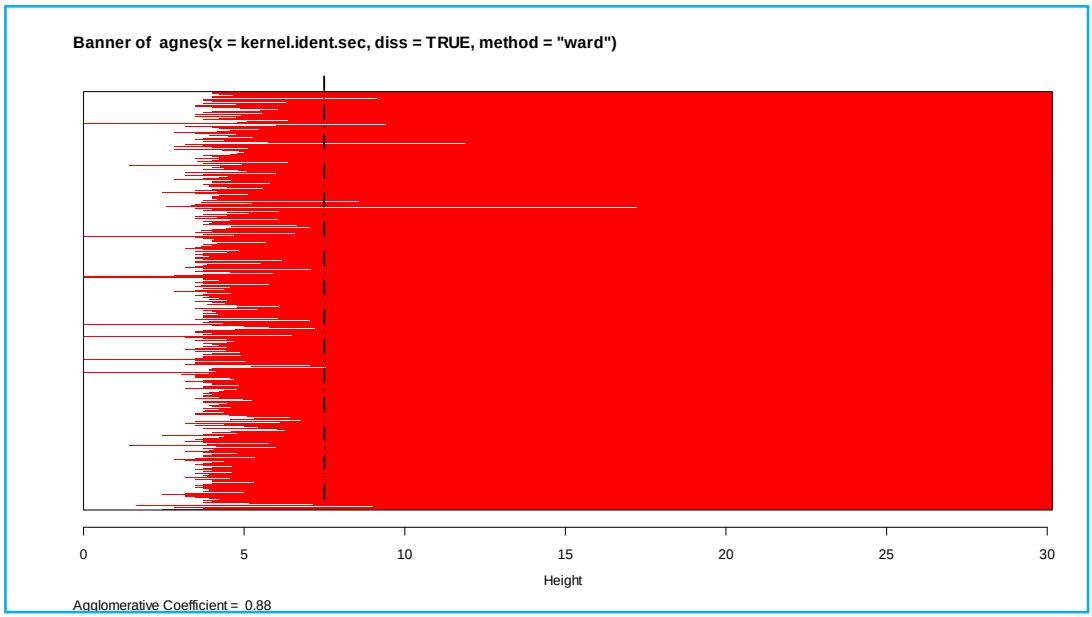
Una vez normalizado el kernel suma, k_{suma}, se aplica el algoritmo AGNES al conjunto de datos con el propósito de obtener el dendograma que permita determinar un parámetro inicial para el algoritmo k-medias. La figura 23 muestra que a una altura de 20 se pueden apreciar 6 grupos de enzimas.

Figura 18. Dendograma obtenido para el kernel suma



Con un coeficiente de aglomeración de 0.88 este caso, resultó ser el de mayor valor obtenido durante las diferentes pruebas, esto indica que la información físico-química introducida en los datos permitió una mejor agrupación de los mismos. La figura 24 muestra el diagrama de barras al aplicar el kernel suma, al trazar una línea vertical a una altura aproximada de 9, se obtuvo 6 cortes con las barras horizontales, lo cual determinó el número de conglomerados a este nivel.

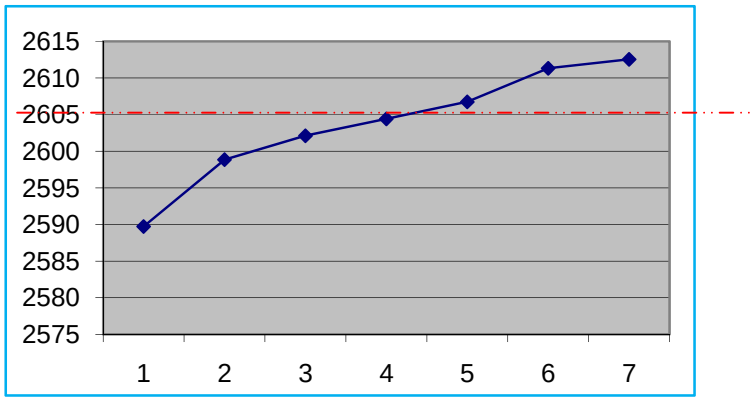
Figura 19. Diagrama de barras con kernel suma



Al igual que en la sección 4.4.1, tomando como referencia el número de grupos que proporcionó el dendograma, se realiza un análisis estadístico referente al número de grupos en los cuales se particionan los datos, considerando para ello diferentes valores del parámetro n(número de grupos) en el algoritmo k-medias.

La figura 25 muestra el error cuadrático medio al considerar el kernel identidad en la aplicación del algoritmo k-medias.

Figura 20. k-medias con kernel suma



En la figura se observa nuevamente que la varianza del error cuadrático es mayor cuando el número de grupos es de cinco grupos.

4.3.3. Interpretación o caracterización de cada Grupo. En el primer grupo está caracterizando las enzimas que contienen el residuo histidina, en forma abreviada, (H, Hys);el cual representa el grupo químico imidazole [36]. La interpretación de este grupo se explica debido a que histidina puede cumplir un papel de catalizador ácido-base [48] ó también de un nucleófilo. Este grupo se conformo con 375 secuencias con residuos catalítico histidina.

El grupo dos estaría formado por enzimas que contienen los residuos arginina (ARG,R) y serina (SER,S).es decir los conformaría el grupo guanidino

determinado por la arginina y el grupo hydroxyl dado por la serina. Este grupo se describe basado en el concepto de unidad catalítica [37], donde se tiene que la interacción arginina- arginina se presenta en cinco enzimas diferentes.Este grupo se conformo con un total de 328 secuencias, de las cuales 201 corresponden al residuo arginina y 126 corresponden al residuo serina.

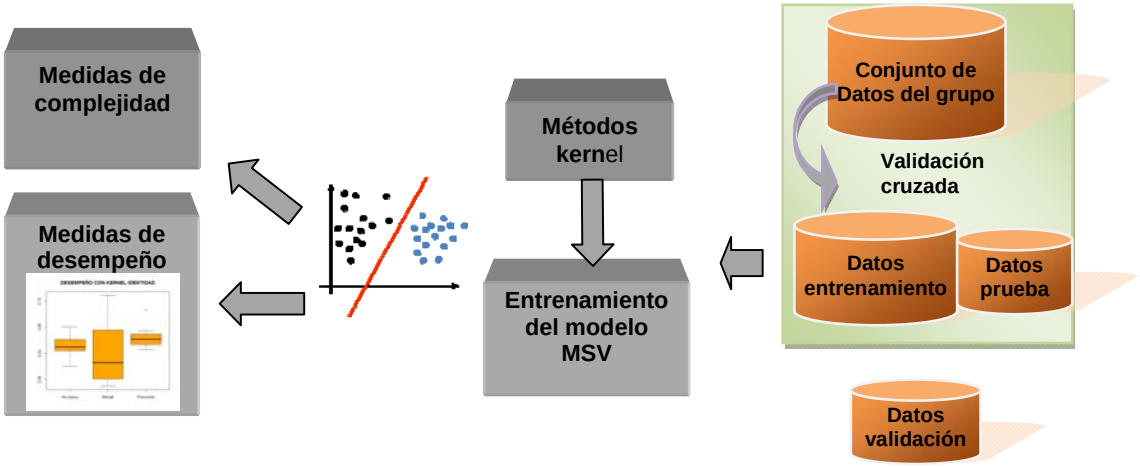
El tercer grupo caracterizado por lisina (LYS,K), cisteína (CYS,C) y asparagina (ASN,N) entre otros residuos. El cuarto grupo conformado por el ácido glutámico (Glu,E) que representa el 11% de los residuos y tirosina (Tyr,Y).
Un quinto grupo está conformado por las secuencias que contiene el residuo catalítico ácido aspártico (D, ASP) asociado al grupo carboxilate [36]. El ácido aspártico se describe como un residuo del tipo carga [8] y por ende tienen valor en la constante de acidez, pKa, que están lejos del valor neutro lo cual hace que no cumplan un papel catalizador en ácido–base. Sin embargo su valor de pKa puede ser alterado de manera significativa y proporcionar cargas que afectan a otros residuos y el sustrato [37].

4.4. Modelo de clasificación propuesto basado en funciones Kernel con MSV

Las etapas siguientes de la metodología propuesta en este capítulo, hace referencia al entrenamiento y validación del modelo de clasificación de residuos catalíticos para cada uno de los grupos determinados en el análisis de conglomerados. La figura 26 muestra en detalle los pasos realizados para la obtención del modelo de un grupo de enzimas.

En la primera etapa los datos de cada grupo de enzimas se dividieron en tres componentes: datos de entrenamiento, datos de prueba y datos de validación del modelo.

Figura 26. Entrenamiento y validación del modelo



Posteriormente, tanto a los datos de entrenamiento, obtenidos como se describió en la sección 3.3, como a los datos de prueba se les aplicó la función kernel polinomial [5], y junto con las MSV permitieron definir una función de clasificación de residuos catalíticos de enzimas.

Así

donde x_i son las secuencias de entrenamiento, y_i son las etiquetas con valor (± 1)

y representa la similitud de las secuencias de enzimas x_i, y, x .

El modelo de clasificación obtenido por esta metodología debe ser evaluado mediante las medidas de desempeño descritas en la sección 1.3.5, al igual que la complejidad del mismo.

4.4.1. Construcción del kernel para la MSV. Cabe recordar que el conjunto de datos está formado por secuencias de enzimas, sin considerar las propiedades fisicoquímicas de las mismas; por ende la construcción del kernel está basado en la teoría de kernels en secuencias, descrita en la sección 2.2, en donde la similitud define el tipo de kernel.

A partir de las propiedades descritas en [46]. Las funciones kernel y las transformaciones permiten definir una amplia gama de kernels más complejos. En esta sección se describen los diferentes kernels utilizados en este trabajo para las subsecuencias de enzimas.

Sean:

X: Espacio de entrada (subsecuencias de enzimas)

x , z: Elementos del espacio de entrada X.

$k: \mathrm{X} \times \mathrm{X} \rightarrow \mathbb{R}$: Función kernel

- El primer kernel utilizado fue el kernel string definido en 2.2.2 como

$$k_p(\mathrm{x}, \mathrm{z}) = \langle \phi^p_{spectrum}(\mathrm{x}), \phi^p_{spectrum}(\mathrm{z}) \rangle$$

Una característica importante del kernel string es que no tiene en cuenta la posición de las subsecuencias de longitud p dentro de la secuencia. Es por ello que para subsecuencias de pequeña longitud, la información respecto a la posición de los residuos catalíticos dentro de la enzima se pierde [39].

- Un segundo kernel utilizado fue el kernel string (full string) dado por

$$k(\mathrm{x},\mathrm{z}) = \langle \phi_{string}(\mathrm{x}), \phi_{string}(\mathrm{z}) \rangle$$

que considera no solamente secuencias de longitud fija p

Se consideró además el kernel identidad [12] definido en la sección 2.2.4. Por último se experimentó con el kernel (RBF) Gaussiano [34], el cual se expresa como

$$k(\mathrm{x},\mathrm{z}) = \exp\left(\frac{-\|\mathrm{x}-\mathrm{z}\|^2}{2\sigma^2}\right)$$

donde $\sigma \in \mathbb{R}^+$

4.4.2. Complejidad computacional del modelo. La complejidad computacional del modelo está asociada con la complejidad de la MSV, el cálculo de los diversos kernels y la complejidad del entrenamiento del modelo. Así

Se denota.

m: número de subsecuencias de enzimas
n: número de vectores soporte
l: longitud de las subsecuencias
$|\mathrm{x}|$: longitud de subsecuencia x
$|\mathrm{z}|$: longitud de las subsecuencia z
$\sum_p$: conjunto de todas las secuencias de longitud p

- La complejidad de una MSV es $O(m^3)$, dado que es un problema de programación cuadrática [45] y el del entrenamiento del clasificador basado en MSV es $O(mn^3)$ [46]
- Complejidad del kernel. En primer lugar, para el kernel string se deben considerar todas las subsecuencias de longitud p, tanto en la enzima x como en z. Luego, se calcula la suma de los productos. Así, se tendría una

complejidad $O(lp)$ [40]. Para el kernel identidad se tendría una complejidad $O(ml)$ [12].

4.4.3. Complejidad en el modelo.

La regularización permite entrenar modelos de tal manera que se disminuya el riesgo de sobre ajuste, por medio del control de la complejidad del modelo

Para la regularización de un modelo se considera la función de error

$$E = \text{error de datos de entrenamiento} + \lambda \, \text{complejidad}$$

donde λ se denomina el parámetro de regularización. Para minimizar el error de generalización se utiliza validación cruzada; Así se utilizan los datos de entrenamiento para variar λ y con datos de validación se mide error de generalización de esta forma se busca estimar la mejor complejidad.

La regularización en MSV se realiza por medio de la constante C [51]. Como se mencionó en la sección 1.3.3 en la MSV se busca el hiperplano que cumpla.

$$\min \frac{1}{2}\|w\| + C\sum_{i=1}^{l} \xi_i$$

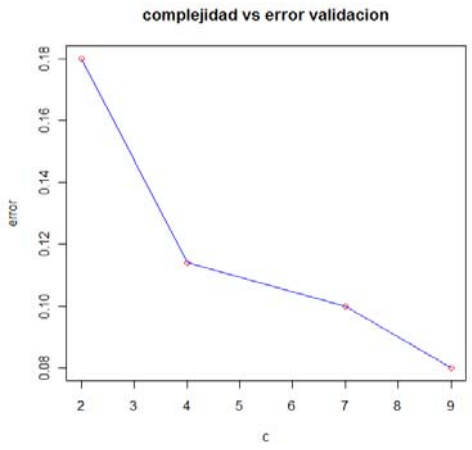
C controla la relación entre la complejidad del MSV y el error de entrenamiento, es decir, controla la penalización por clasificar mal un elemento del conjunto de entrenamiento. Un valor alto de C hace que la MSV genere una función de predicción bastante compleja, la cual clasifica mal unos pocos elementos del conjunto de entrenamiento y por el contrario un valor bajo de C genera una función de predicción sencilla [49].

En la práctica, para escoger el valor más adecuado del parámetro C, se utilizó el conjunto de entrenamiento de la técnica de validación cruzada variando C en un rango de valores y se verifica el desempeño con el conjunto de validación por medio del error de generalización [50].

En la figura 27 se aprecia que a medida que aumenta la complejidad del modelo, el error del conjunto de validación disminuye hasta un cierto nivel de complejidad, y se detiene o no disminuye aún más, o incluso aumenta si hay ruido significativo. Este cambio ó "codo" corresponde a la complejidad nivel óptimo [15]

La figura 27 presenta el error para diversos valores de C, en la cual se aprecia que la mejor complejidad se tiene cuando C es aproximadamente 4. Cabe notar que el kernel que se usó fue polinomial obtenido con un polinomio de grado dos y el kernel string.

Figura 27. Complejidad del modelo



4.5. Detalles de implementación

Para cada una de las etapas de la metodología propuesta en el capítulo cuatro se hizo necesario implementar rutinas, ya sea en Matlab [42] o en el paquete estadístico de software libre R [43].

Para la etapa de preprocesamiento se implementaron algoritmos para formar un solo archivo de datos ya que los datos originales tomados del sitio oficial de catalytic sites atlas contiene las enzimas en archivos independientes.

En la etapa de agrupamiento se utilizó la librería cluster de R. Para la validación y entrenamiento del modelo con los kernels string se usó la librería Kernlab de R [44], y de igual forma para los kernels RBF y los kernels con propiedades físico-químicos se implementaron los algoritmos en Matlab, en especial para obtener la matriz de kernel. Para la presentación y el cálculo de las medidas de desempeño se utilizó tanto R como Matlab.

5. Validación experimental del modelo

Uno de los elementos en la generalización de un modelo hace referencia a la complejidad del mismo, es por ello que para obtener un modelo con la mejor complejidad se utilizó la técnica de validación cruzada. El conjunto de datos se dividió en tres partes; un conjunto de entrenamiento, uno de prueba y por último un conjunto de validación. Con los dos primeros conjuntos se entrenan modelos obtenidos al variar la complejidad en un rango de valores, y medir así los errores de entrenamiento y de prueba obtenidos en los diferentes modelos. El conjunto de validación permite, como su nombre lo indica, validar los resultados obtenidos y determinar de esta manera la mejor complejidad. Esto con el fin de ajustar los parámetros que permitan obtener un modelo con desempeño más alto al obtenido en la sección 3.3.1.

En este capítulo se analiza el desempeño obtenido por el clasificador en cada uno de los grupos variando el parámetro de complejidad del modelo y los parámetros n y λ de cada kernel.

5.1. Análisis de resultados variando el parámetro *C* en el Kernel identidad

Para el primer grupo caracterizado por la histidina, se varió el parámetro C, definido en la sección 4.4.3, para los valores {2,4,6,8,10} ya que para valores superiores de C las secuencias se clasifican en un sola clase, al igual que sucedió en la experimentación inicial de los datos, sección 3.4.1.

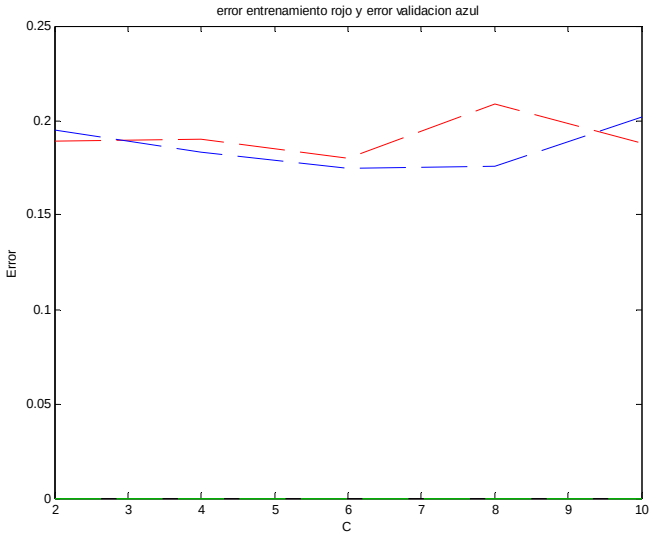
Con validación cruzada y el promedio de 15 iteraciones realizadas se obtuvieron los errores promedio de entrenamiento, prueba y validación; presentados en la tabla 8.

Tabla 5. Errores promedio con kernel identidad para el grupo 1.

Kernel IDENTIDAD	C=2	C=4	C=6	C=8	C=10
Error de Entrenamiento	0	0	0	0	0
Error de prueba	0,189	0,190	0,180	0,209	0,188
Error de validación	0,195	0,183	0,176	0,176	0,202

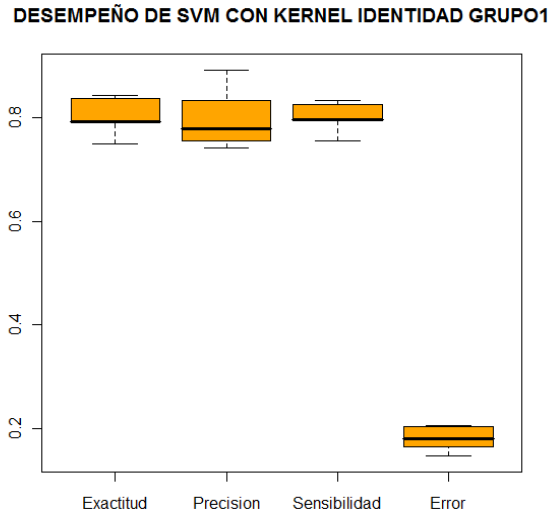
En la gráfica 28 se muestra el comportamiento de dichos errores al variar C. En ella se aprecia que a medida que aumenta la complejidad del modelo, el error de entrenamiento, color rojo, disminuyen al igual que el error en el conjunto de validación, color azul, este último disminuye hasta un cierto nivel de complejidad, C=6, y se detiene o no disminuye aún más, o incluso aumenta si hay ruido significativo [15]. Este cambio o "codo" corresponde a la complejidad nivel óptimo.

Figura 28. Complejidad vs error grupo 1 con kernel identidad



En la figura 29 se muestran las medidas de desempeños para C=6

Figura 29. Medidas de desempeño con kernel identidad para grupo 1.



Para este grupo se tuvo una exactitud de 81% y un error aproximado del 18%

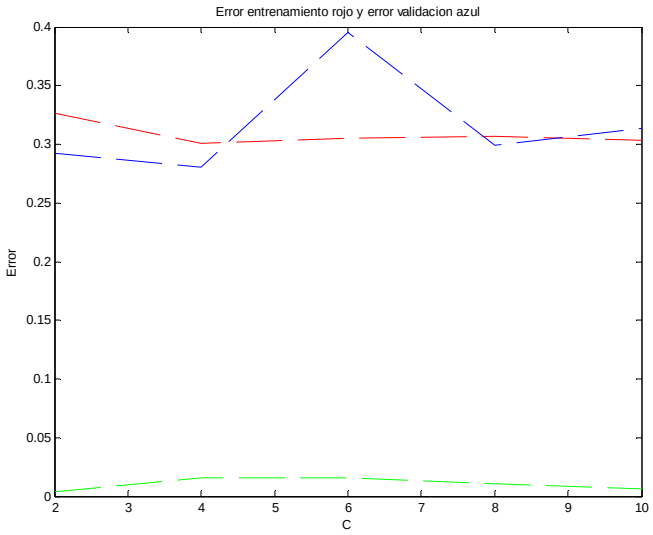
Para el segundo grupo caracterizado arginina (ARG,R) y serina (SER,S) se obtuvieron los errores de validación dados por la tabla 9.

Tabla 9. Errores promedio con kernel identidad para grupo 2.

Kernel IDENTIDAD	C=2	C=4	C=6	C=8	C=10
Error de Entrenamiento	0,004	0,016	0,016	0,011	0,006
Error de prueba	0,326	0,301	0,305	0,3076	0,303
Error de validación	0,295	0,280	0,395	0,299	0,314

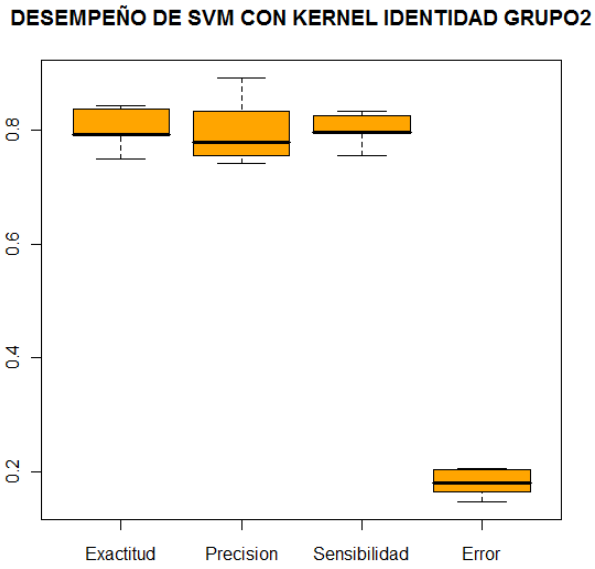
En esta tabla se muestra que el menor error de validación se tiene para C igual a 4. Este valor que se confirma al graficar dichos errores, como se aprecia en la figura 30 y al calcular el desempeño para los diferentes valores de C.

Figura 21. Complejidad vs error grupo 2 con kernel identidad



Al realizar 15 iteraciones para el modelo con el parámetro C igual a cuatro se obtuvieron las siguientes medidas de desempeño.

Figura 22. Medidas de desempeño con kernel identidad para grupo 2



La figura 31 muestra una exactitud del 80% y un error del 20%, al igual una precisión aproximada del 79%.

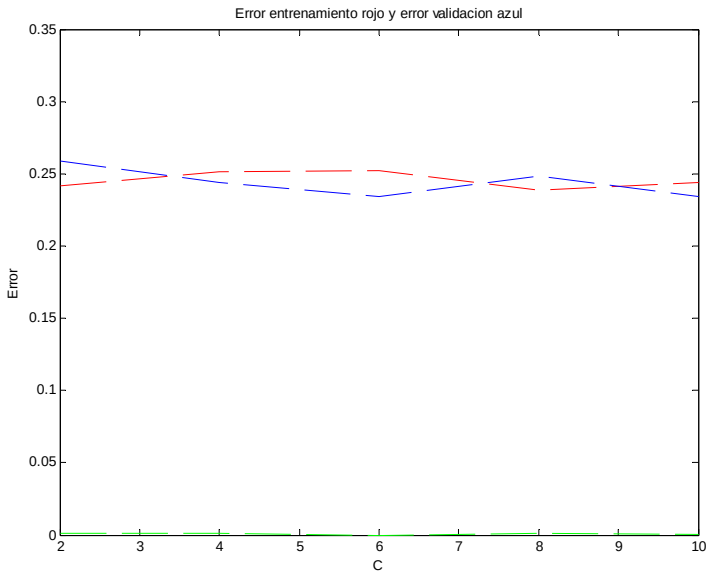
Para el tercer grupo caracterizado por lisina (Lys,K), cisteína (Cys,C) y asparagina (ASN,N) entre otros, se obtuvieron los errores que se aprecian en la tabla 10

Tabla 6. Errores promedio con kernel identidad para grupo 3

Kernel Identidad	C=2	C=4	C=6	C=8	C=10
Error de entrenamiento	0	0	0	0	0
Error de prueba	0,242	0,251	0,252	0,239	0,244
Error de validación	0,259	0,244	0,234	0,248	0,234

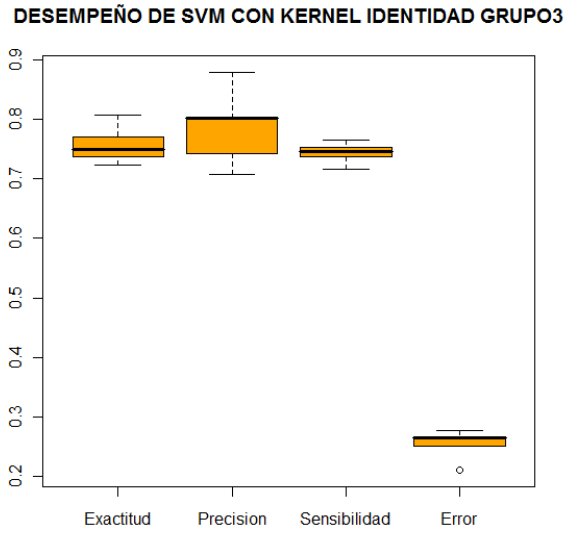
Estos errores muestran que el valor de C que tiene el mínimo error de validación es de seis como se aprecia en la figura 32

Figura 23. Complejidad vs error grupo 3 con kernel identidad



La exactitud aproximada del clasificador para este grupo fue del 75%, el cual representa el desempeño más bajo que se tiene hasta ahora. El error que se obtuvo fue de 25% aproximadamente. Como se aprecia en la figura 33.

Figura 24. Medidas de desempeño con kernel identidad para grupo 3

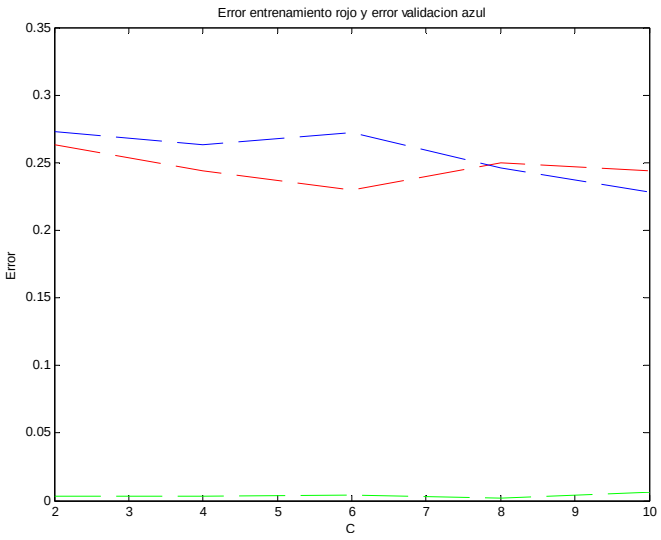


Para el cuarto grupo conformado por el ácido glutámico (Glu,E) y tirosina (Tyr,Y), los errores de validación obtenidos en las 15 iteraciones se muestran en la tabla 11, al igual la gráfica de los mismos se presenta en la figura 34.

Tabla 7. Errores promedio con kernel identidad para grupo 4

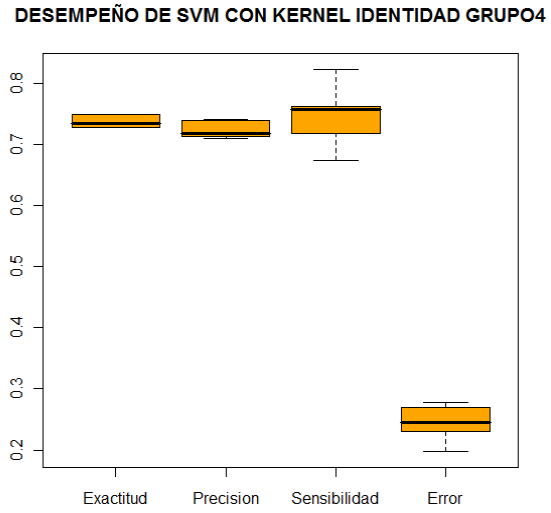
Kernel identidad	C=2	C=4	C=6	C=8	C=10
Error de Entrenamiento	0,003	0,003	0,004	0,002	0,006
Error de prueba	0,263	0,244	0,23	0,25	0,244
Error de validación	0,273	0,263	0,273	0,246	0,228

Figura 25. Complejidad vs error grupo 4 con kernel identidad.



En la gráfica anterior se aprecia que en C igual a ocho el error es menor, con este parámetro se tiene una exactitud del 74% aproximadamente y un error aproximado del 25%, como se ve en la figura 35.

Figura 26. Medidas de desempeño con kernel identidad para grupo 4

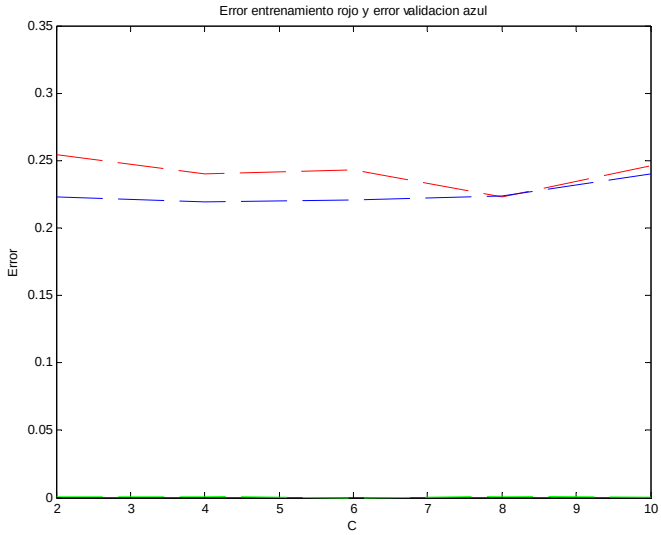


Por último, para el kernel identidad con el quinto grupo, dado por el ácido aspártico (Asp,D), se tiene que el mínimo error de validación se presenta en el parámetro C igual a cuatro como se aprecia en la tabla 12 y se verifica en la figura 36

Tabla 12. Errores promedio con kernel identidad para grupo 5

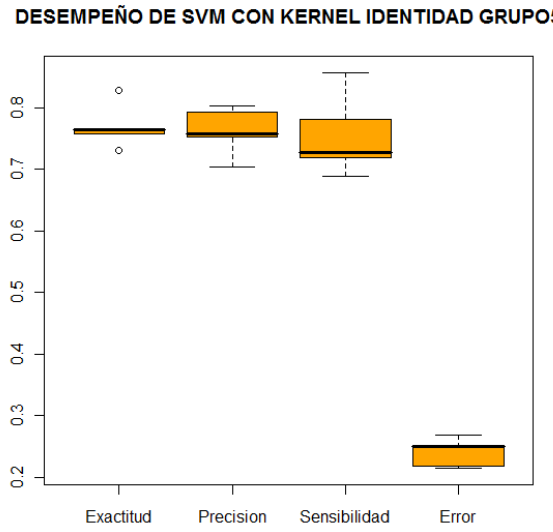
Kernel identidad	C=2	C=4	C=6	C=8	C=10
Error de Entrenamiento	0,001	0,001	0	0,001	0,0005
Error de prueba	0,254	0,240	0,243	0,223	0,246
Error de validación	0,223	0,219	0,221	0,224	0,24

Figura 36. Complejidad vs error grupo 5 con kernel identidad



El error menor de validación y entrenamiento se incrementa para valores de C superiores a ocho. En la figura 37 se presenta la exactitud y las demás medidas de desempeño.

Figura 37. Medidas de desempeño con kernel identidad para grupo 5



Se aprecia que la exactitud es aproximadamente del 77% con una varianza mínima y además se tiene un error del 25% aproximadamente.

5.2. Análisis de resultados variando los parámetros C y n en el Kernel String.

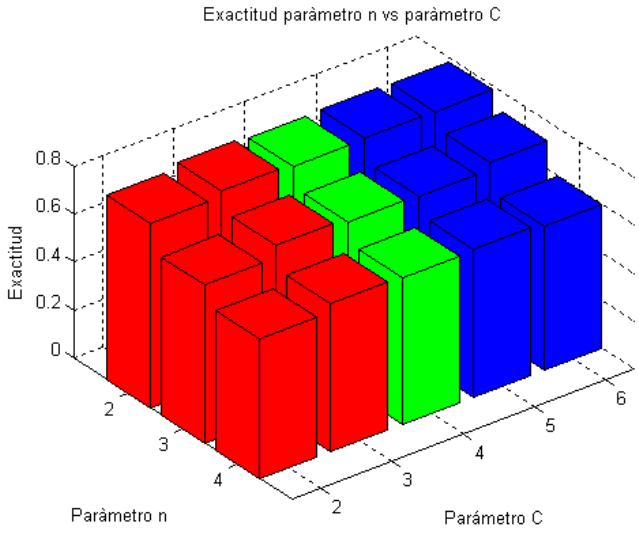
En esta sección al igual que con el kernel identidad se utilizó la técnica de validación cruzada junto con las medidas de desempeño definidas en la sección 1.3.5. Para tal validación se variaron cada uno de los parámetros del modelo y se calcula el desempeño del modelo.

Uno de los parámetros del kernel string es factor de decadencia λ, definido en la sección 2.2, y cuyo valor se fijó en 0.5 ya que en la experimentación de la sección 3.4.1 se verifico que ese valor de λ es el que permite un mejor desempeño para el clasificador. Es por ello que en esta sección se varían solamente la longitud de las subsecuencias, parámetro n, y el parámetro C, para de esta manera, obtener el modelo con el mejor desempeño posible en cada grupo. Cabe recordar que en

dicha experimentación se encontró que los valores adecuados de n están entre dos y cuatro.

La figura 38 presenta el promedio de la exactitud al realizar 15 iteraciones para el primer grupo variando el parámetro n entre el conjunto de valores {2,3,4} y el parámetro C en el conjunto { 2,3,4,5,6}

Figura 27. Exactitud variando los parámetros n y C en grupo 1

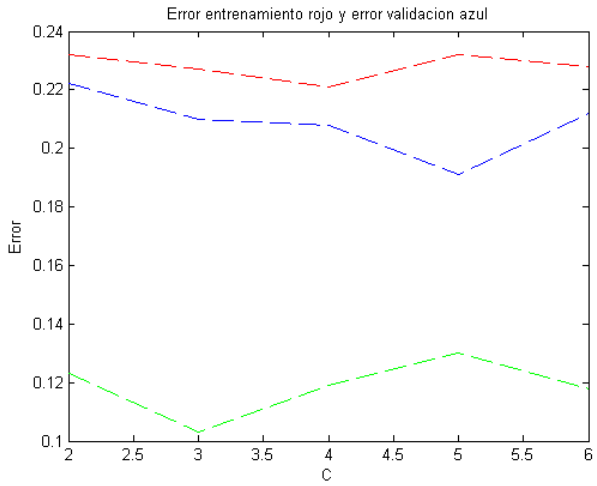


En la anterior gráfica se aprecia que los desempeños más altos están cuando n toma el valor de dos. Es por ello que para encontrar el valor adecuado del parámetro C se graficaron los errores promedio de las 15 iteraciones realizadas y se resumen en la tabla 13 y se presenta en la figura 39

Tabla 8. Errores promedio variando C en kernel string para grupo 1

n=2	C=2	C=3	C=4	C=5	C=6
Error entrenamiento	0,123	0,103	0,119	0,130	0,118
Error prueba	0,232	0,227	0,221	0,232	0,228
Error validación	0,222	0,21	0,208	0,191	0,212

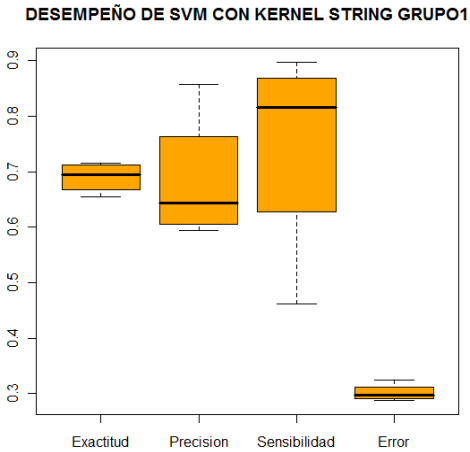
Figura 39. Complejidad vs error grupo 1 con kernel string



En la anterior figura se aprecia que el mínimo error de validación se tiene cuando el parámetro C toma valor de cinco, sin embargo el error de entrenamiento es menor para un valor de C igual a cuatro. Teniendo presente que el conjunto de validación no hace parte de los datos de entrenamiento con un valor de C=5 se tendrá el mejor desempeño.

La figura 40 muestra las medidas de desempeño para el clasificador con el valor fijado para dicho parámetro.

Figura 28. Medidas de desempeño con kernel string para grupo 1



La exactitud obtenida por el clasificador con los parámetros n igual a dos C igual a cinco fue del 70% y el error del 30%.

Para el grupo dos se observa en la figura 41 que el mejor desempeño se tiene con el parámetro n igual a dos; para verificar en qué valor del parámetro C se tiene el menor error se graficaron los errores de entrenamiento y validación dados en la tabla 14.

Figura 29. Exactitud variando los parámetros n y C en grupo 2.

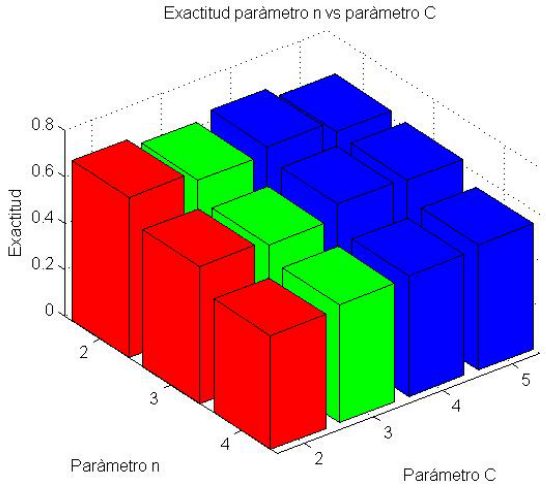
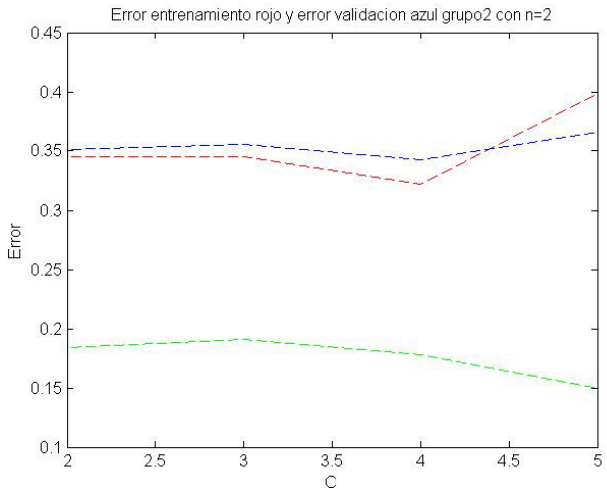


Tabla 9. Errores promedio variando C en kernel string para grupo 2

n=2	C=2	C=3	C=4	C=5
Error de entrenamiento	0,184	0,191	0,178	0,150
Error de prueba	0,345	0,345	0,322	0,399
Error de validación	0,351	0,356	0,343	0,366

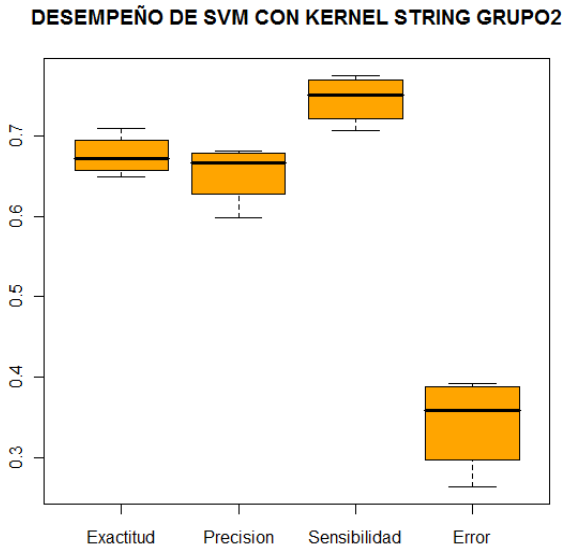
En la figura 42 se observa que la complejidad que minimiza el error se encuentra con el valor de C igual a cuatro. Para valores superiores a cuatro, tanto el error de entrenamiento como de validación se incrementan.

Figura 30. Complejidad vs error grupo 2



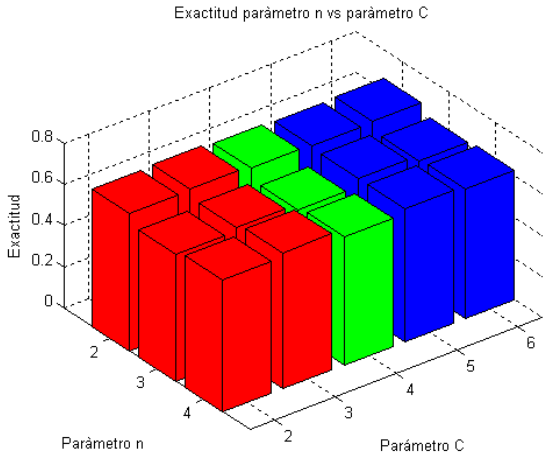
Las medidas de desempeño para los parámetros n igual a dos y C igual a cuatro se muestran en la figura 43. Se aprecia que la exactitud está cerca del 68% al igual que la precisión; y el error en promedio que se tiene es aproximadamente del 35%.

Figura 31. Medidas de desempeño con kernel string para grupo 2



De forma análoga, para el grupo tres se tiene que el mejor valor de la exactitud se tiene para el parámetro n igual a dos como se aprecia en las figuras 44.

Figura 32. Exactitud variando los parámetros n y C en grupo 3.

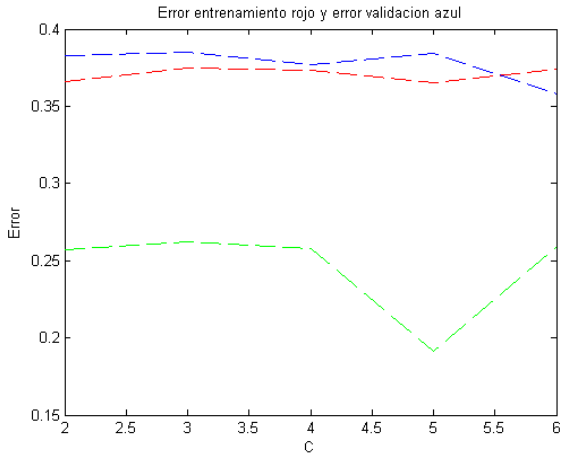


El error de validación menor se tiene para el parámetro C igual cuatro como se aprecia en la tabla 15 y en la figura 45. También se aprecia que el error de entrenamiento es mayor para este grupo.

Tabla 10. Errores promedio variando C en kernel string para grupo 3

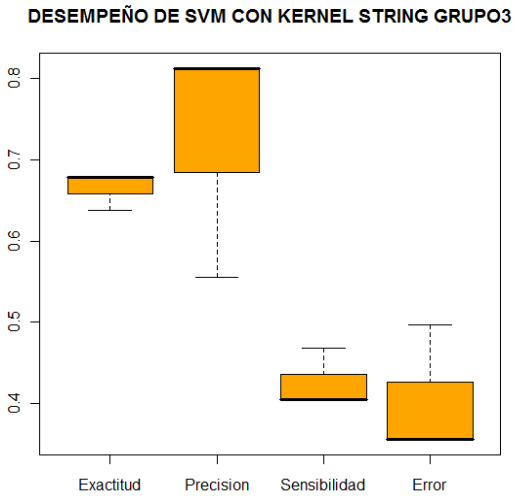
n=2	C=2	C=3	C=4	C=5	C=6
Error de entrenamiento	0,257	0,262	0,258	0,191	0,259
Error de prueba	0,366	0,375	0,373	0,365	0,374
Error de validación	0,383	0,385	0,377	0,384	0,358

Figura 11. Complejidad vs error grupo 3



Las medidas de desempeño del clasificador con parámetros n igual a dos y C igual a cuatro para este grupo muestran que la exactitud promedio es del 68% y un error por encima del 35%.

Figura 33. Medidas de desempeño con kernel string para grupo 3



En el grupo cuatro se tiene que el mayor desempeño se tiene entre los valores de C igual a cuatro y C igual a dos, como se aprecia en la figura 47, es por ello que se hace necesario graficar los errores de entrenamiento y validación, dados en la

tabla 16, para así poder determinar el valor del parámetro C que proporcione mejor desempeño.

Figura 47. Exactitud variando los parámetros n y C en grupo 4.

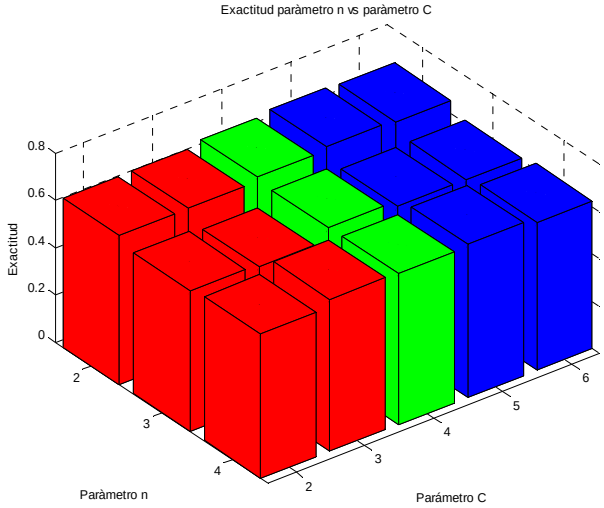
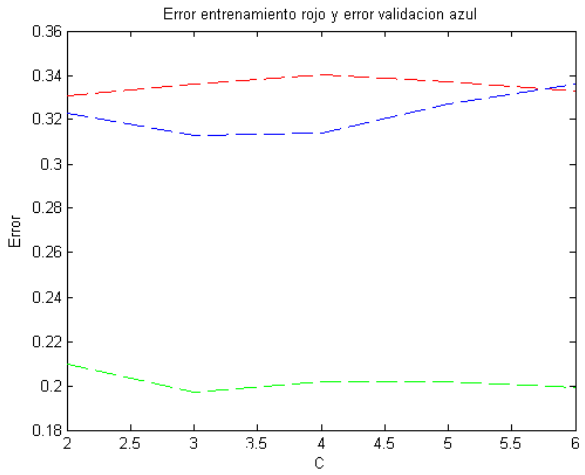


Tabla 16. Errores promedio variando C en kernel string para grupo 4

n=2	C=2	C=3	C=4	C=5	C=6
Error de entrenamiento	0,210	0,197	0,202	0,202	0,199
Error de prueba	0,331	0,336	0,340	0,337	0,333
Error de validación	0,323	0,313	0,314	0,327	0,336

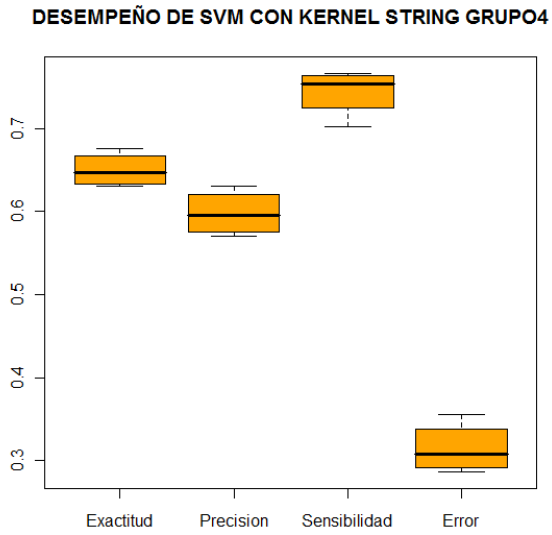
Al analizar la gráfica de la figura 48 se aprecia que el menor error de validación se tiene para C igual a cuatro, ya que el promedio general para C igual a dos es superior al 30%.

Figura 48. Complejidad vs error grupo 4



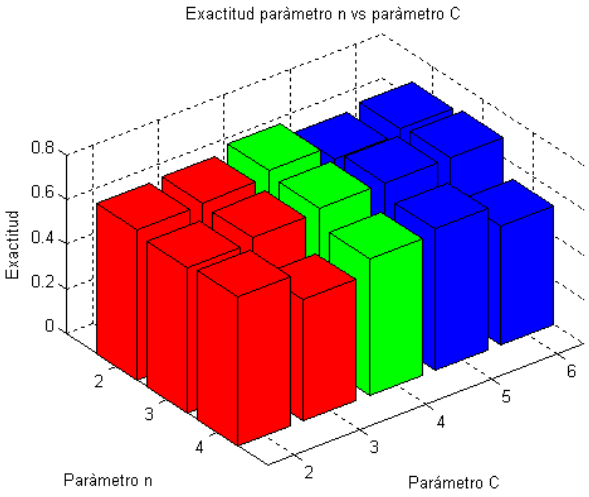
Los niveles de desempeño para la MSV con los parámetros n igual a 2 y el parámetro C igual a cuatro se muestran en la figura 49, en la gráfica se verifica que la exactitud está cerca al 65% y el error en un 30%.

Figura 49. Medidas de desempeño con kernel string para grupo 4.



Por último, el grupo cinco caracterizado por el ácido aspártico (Asp,D), logra un mejor desempeño para n igual a 3, como se puede observar en la figura 50.

Figura 34. Exactitud variando los parámetros n y C en grupo 5

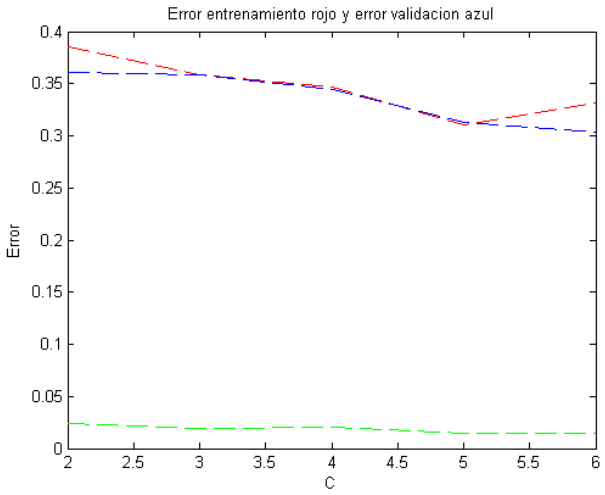


Al revisar en la tabla 17, los errores promedio de entrenamiento y validación, obtenidos al realizar las 15 iteraciones, se puede apreciar que el error de validación menor para el parámetro C con un valor de cinco es de 31% aproximadamente. Aunque para C igual a seis se tiene un error menor el parámetro con dicho valor no se considera pues en la figura 51 se aprecia que el "codo" se tiene en el valor de cinco para C.

Tabla 17. Errores promedio variando C en kernel string para grupo 5

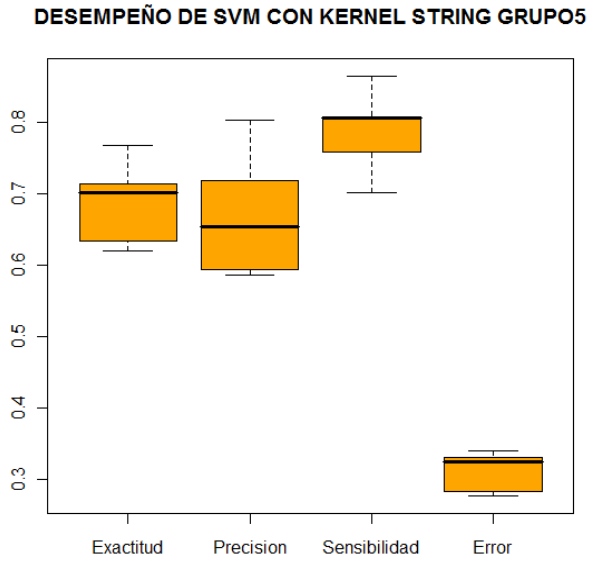
n=3	C=2	C=3	C=4	C=5	C=6
Error de entrenamiento	0,024	0,019	0,020	0,015	0,015
Error de prueba	0,385	0,359	0,347	0,311	0,331
Error de validación	0,361	0,359	0,344	0,313	0,304

Figura 35. Complejidad vs error grupo 5



Las medidas de desempeño para este grupo muestran que la exactitud es del 70% y el error está por encima del 30%.

Figura 36. Medidas de desempeño con kernel string para grupo 5.



5.3. Comparación de resultados con otros trabajos

Para la comparación de los resultados obtenidos en este trabajo con otros del mismo enfoque, donde se apliquen kernels string a datos de tipo secuencias proteicas ó genómicas y que además utilicen como clasificador MSV´s, es necesario tener en cuenta los siguientes escenarios.

Uno de ellos conformado por los trabajos propuestos que representan los datos en un espacio de características de dimensión fija (vector de características), utilizan MSV y kernels pero no del tipo string, como en los trabajos de Petrova [8], Sunshin [10], Leslie [40], Bock [2]. Otro escenario lo determinan los trabajos donde el conjunto de datos son del tipo genómico, como por ejemplo los trabajos de Bing [1] y Terribilini [21].

Sin embargo al comparar los resultados obtenidos en trabajo de Wu [12] en donde utilizan las mismas medidas de desempeño y las MSV con kernels string se observa que la metodología y el modelo propuesto en este trabajo mejoran los desempeños del mismo; como se aprecia al revisar los resultados publicados en la tabla 18.

Tabla 18. Resultados publicados de comparación de kernels

TABLE I

COMPARISON OF THE AMINO ACID IDENTITY KERNEL $\mathbf{K}_i$, THE ALIGNMENT KERNEL $\mathbf{K}_a$, AND SEVERAL SUBSTITUTION KERNELS $\mathbf{K}_{sh}$ $\mathbf{K}_{sj}$ AND $\mathbf{K}_{sm}$ (DERIVED FROM HENS920102, JOHM930101, AND MCLA720101 SUBSTITUTION MATRICES RESPECTIVELY). ACCURACY (*ac*), RECALL(*re*), PRECISION (*pr*), AND CORRELATION COEFFICIENT (*cc*) SHOWN ARE ESTIMATED USING LEAVE-ONE-OUT CROSS-VALIDATION.

data set	kernel function	*ac*	*re*	*pr*	*cc*
P	$\mathbf{K}_i$	60.3%	54.9%	42.0%	16.6%
	$\mathbf{K}_a$	63.7%	47.6%	43.9%	16.6%
	$\mathbf{K}_{sh}$	63.4%	48.1%	44.0%	17.7%
	$\mathbf{K}_{sj}$	63.6%	49.7%	44.5%	**18.9%**
	$\mathbf{K}_{sm}$	62.0%	51.4%	42.7%	17.0%
D	$\mathbf{K}_i$	64.0%	69.6%	30.0%	25.0%
	$\mathbf{K}_a$	63.9%	66.0%	29.4%	22.7%
	$\mathbf{K}_{sh}$	64.1%	69.3%	29.7%	24.4%
	$\mathbf{K}_{sj}$	64.4%	68.1%	29.8%	24.3%
	$\mathbf{K}_{sm}$	65.1%	69.6%	30.3%	**25.7%**
R	$\mathbf{K}_i$	71.2%	60.3%	34.8%	25.1%
	$\mathbf{K}_a$	69.2%	53.1%	31.9%	18.0%
	$\mathbf{K}_{sh}$	72.1%	58.4%	35.3%	24.9%
	$\mathbf{K}_{sj}$	72.2%	58.9%	35.5%	**25.3%**
	$\mathbf{K}_{sm}$	71.6%	58.6%	34.8%	24.3%

En la tabla 19 se aprecia que la exactitud está comprendido entre el 60% y el 72% para tres diferentes conjuntos de datos y con cinco funciones kernels diferentes.

En la sección 3.3.1 se presentaron los resultados de realizar una experimentación inicial sobre los datos; en dicha experimentación se logro un desempeño del 65% de exactitud al aplicar el kernel identidad y un desempeño del 60% de exactitud con el kernel string. Los resultados obtenidos son cercanos a los reportados por Wu [12]

Al aplicar la metodología del capítulo 4, se obtuvo con el kernel identidad un desempeño comprendido entre el 75% y 80% de exactitud y con el kernel string una exactitud entre del 65% y 71%, como se aprecia en la tabla 19. Lo cual muestra que el desempeño en general es superior al reportado en la tabla 18.

Tabla 19. Medidas de desempeño en cada grupo con kernel identidad y kernel string

Grupo	Kernel	Exactitud	Precisión	Sensibilidad
1	String	70%	65%	83%
	Identidad	81%	79%	80%
2	String	68%	67%	75%
	Identidad	80%	78%	80%
3	String	69%	81%	47%
	Identidad	75%	80%	75%
4	String	65%	59%	75%
	Identidad	75%	74%	76%
5	String	70%	65%	80%
	Identidad	75%	74%	73%

Conclusiones y trabajo futuro

Una de las etapas más dispendiosas y relevantes al aplicar los métodos de aprendizaje en un sistema de clasificación para datos del tipo secuencia biológica, hace referencia al preprocesamiento de los mismos. Como en el caso de las enzimas en donde los residuos catalíticos que determinan un sitio activo no se encuentran cercanos en secuencia y el hecho que la mayoría de las enzimas tiene de uno a cinco residuos que intervienen en la catálisis**,** es adecuado realizar una representación de los datos, en la cual no se considera la totalidad de la secuencia debido a que se esto genera mayor error en el proceso de predicción de los residuos catalíticos.

En el preprocesamiento de los datos es necesario tener presente las enzimas que contienen triadas catalíticas, ya que estos residuos aportan información biológica a la conformación de los grupos.

Las propiedades de clausura permite definir kernels más complejos o combinar funciones kernel de diferentes fuentes de datos. Por ello es necesario normalizar el kernel obtenido al aplicar dichas propiedades esto para evitar el sesgo producido por la diferente naturaleza de los datos de entrada. En este trabajo el coeficiente de aglomeración obtenido al aplicar las propiedad de suma de los kernels obtenidos por la información de la secuencia y el obtenido por la información físico-química de los residuos resultó ser mayor que el coeficiente considerando solamente la secuencia, esto, indica que la información de las anotaciones introducida en los datos permitió una mejor agrupación de los mismos.

Con la experimentación realizada en este trabajo referente al modelo de clasificación de las secuencias de enzimas, el factor de decadencia para los kernels string, sequence, stringy full string no afecta significativamente el

desempeño del modelo, por esta razón se puede realizar la experimentación fijando su valor.

Al comparar el desempeño de los kernels identidad, kernel string, sequence, string y full string, el mejor desempeño tanto en la etapa de conformación de grupos como en la de entrenamiento y validación del modelo, fue obtenido usando el kernel identidad, el siguiente kernel en lograr mejor desempeño fue el kernel string.

Con los resultados obtenidos en la experimentación se puede apreciar que la metodología propuesta al incluir técnicas de agrupación, mejoró el desempeño del clasificador para la predicción de residuos catalíticos en enzimas.

Una de las etapas de la metodología que presentó algún tipo de inconveniente fue precisamente el análisis de agrupación, en especial en el proceso de caracterizar cada grupo desde un punto de vista biológico.

Se pretende realizar un futuro trabajo orientado a predecir sitios de unión de interacciones proteína-proteína, ARN-proteína considerando la estructura primaria junto con las anotaciones reportadas que se obtenga de la estructura 3D. Esto motivado por el conocimiento de los mecanismos de regulación de la expresión proteica a nivel celular.

Por último, el crecimiento de las bases de datos de proteínas de las cuales no se conoce su función y el alto costo que requiere la determinación de las funciones de las mismas, así como los mecanismos desencadenados para llevar a cabo tales funciones, permite pensar en realizar un posterior trabajo orientado a proponer un de predicción de la función de una proteína a partir de la secuencia y de la información físico-química que se obtiene en la estructura 3D.

Bibliografía.

[1] Bing Wang, Hau San Wong, Peng Chen, Predicting Protein-Protein Interaction Sites Using Radial Basis Function Neural Networks, 2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006,

[2] Bock, J. R. & Gough, D. A. Predicting protein-protein interactions from primary structure (2001) BIOINFORMATICS

[3] Fariselli, P., Pazos, F., Valencia, A. & Casadio, R. Prediction of protein protein interaction sites in hetero- complexes with neural networks (2002) European journal of biochemistry / FEBS

[4] HH Lin1, L. H. Prediction of the functional class of metal-binding proteins from sequence derived physicochemical properties by support vector machine approach (2006) BMC Bioinformatics

[5] John Shawe-Taylor & Nello Cristianini. Kernel Methods for Pattern Analysis.Cambridge University Press, 2004

[6] Mount, David, Bioinformatics: Sequence and Genome Analysis, 2001, Cold Spring Harbor Laboratory Press,pg 390,392

[7] N. Cristianini y J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, 2000

[8] Petrova, N. V. & Wu, C. H. Prediction of catalytic residues using Support Vector Machine with selected protein sequence and structural properties 2006 . BMC bioinformatics, pg 8

[9] Saigo, H., Vert, J., Ueda, N. & Akutsu, T. Protein homology detection using string alignment kernels 2004 Bioinformatics (Oxford, England)

[10] Sunshin Kim and Chung Rhee. A New Algorithm to Predict the Active Sites Using Amino Acid Vectors and Biochemical features of Surface Patches Titulo BIOCOMP 2006.pg 367-372

[11] Vapnik, V.: The natural of statistical Learning Theory. Springer-Verlag, New York (1995).

[12] Wu, F., Olson, B., Dobbs, D. & Honavar, V. Comparing Kernels for Predicting Protein Binding Sites from Amino Acid Sequence.2006 Neural Networks 2006. IJCNN '06. International Joint Conference onNeural Networks, 2006. IJCNN

[13] Wu, X.; Wang, J. T. L. & Herbert. K. G. A New Kernel Method for RNA Classification BioInformatics and BioEngineering. 2006. Sixth IEEE Symposium

[14] Curtis Helena and Barnes N,Biología.Editorial Médica Panamericana. Quinta edicion.1993

[15] Alpaydin, E. Introduction to Machine Learning . The MIT Press.2004 pg3.

[16] B.Alberts, A.Johnson, J.Lewis, M.Ra, K.Roberts, and P.Walter. Molecular Biology of the Cell. 4th ed. New York: Garland Publishing, 2002.

[17] Yan et al. 2004 a,b, 2006.

[18] Curtis Helena and Barnes N,Biología.Editorial Medica Panamericana. Quinta edición.1993

[19] Tuo zhan, hua zhan. Accurate sequence-based predict of catalytic residues.Bioinformatics vol. 24 n° 20 2008 pg 2329-2338

[20] Changhui Yan, Vasant Honavar. Technical Report ISU-CS-TR-02-11.Department of Computer Science. Iowa State University. October 2002

[21] Terribilini Michael, Jae Hyung. Prediction of RNA binding sites in proteins from aminoacid sequence. RNA journal.2006,12

[22] Mehemd Kantardzic. Data mining.Editorial IEEE Press.2003, pag 86

[23] An introduction to cluster analysisKaufman and Rousseeuw,Editorial Wiley.1990

[24] David Lewis. Reuters-21578 text categorization test collection, 1997.

[25] Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using stringkernels. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, Journal of Machine Learning Research 2 (2002) 419-444.

[26] Karatzoglou, Alexandros; Feinerer, Ingo: Text Clustering with String Kernels in R .Research Report Series / Department of Statistics and Mathematics, Nr. 34, May 2006, GfKl 2006, Berlin, Germany

[27] Jaz Kandola,Nello Cristianini, Andre Elisseeff, John Shawe-Taylor .On kernel-target alignment (2002) by Advances in Neural Information Processing Systems 14 pg 367-373. Cambridge, 2002.MIT Press.

[28] Ramazan Türkmen and Zübeyde Ulukök, Journal of Inequalities and Applications.Volume 2010 (2010), Article ID 897279, 11 pages. Research Article . On The Frobenius Condition Number of Positive Definite Matrices.

[29] Chothia, C. (1976). "The nature of the accessible and buried surfaces in proteins." J. Mol. Biol.105: 1-14.

[30] A.A. Zamyatin, Protein Volume in Solution, Prog. Biophys. Mol. Biol. 24(1972)

[31] http://www.imb-jena.de. The Amino Acid Repository, JENA lybrary(página de información físico química).

[32] Pavlidis, P., et al., Gene functional classification from heterogeneous data.
Journal of Computational Biology, 2002. 9(2): p. 401-411

[33] Ben-Hur, A. and W.S. Noble, Kernel method for predicting protein-protein interactions. Bioinformatics, 2005. 21(Suppl. 1): p. i38-i46.

[34] Hsu, C.-W., C.-C. Chang, and C.-J. Lin, A practical guide to support vector classification, in Technical report. 2003, Department of Computer Science and Information Engineering, National Taiwan University

[35] Scholkopf, B., K. Tsuda, and J.P. Vert, Kernel Methods in Computational Biology. 2004: The MIT Press.

[36] Gutteridge. A and Thornton J. Understanding nature´s catalytic toolkit

[37] Gail J. Bartlet,Analysis of Catalytic Residues in Enzyme Active Sites

[38] Bobadilla L and Niño F.Characterizing and Predicting Catalytic Residues in Enzyme Active Sites Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference

[39] Lynne Davis, John Hawkins. Comparing MSV sequence kernels. Proceeding WISB '06 Proceedings of the 2006 workshop on Intelligent systems for bioinformatics. Vol 73 Pages 39-47.

[40] leslie C, Eskin E. the string kernel: a string kernel for MSV protein classification. Pacific Symposium on Biocomputing,2002

[41] Ruth M. Saecker,Tom Record .Kinetic Studies and Structural Models of the Association of E. coli s70 RNA Polymerase with the λPR Promoter: Large Scale Conformational Changes in Forming the Kinetically Significant Intermediates. J. Mol. Biol. (2002) 319, 649–671

[42] Matlab7. http://www.mathworks.com/products/matlab/

[43] Software libre R .http://www.r-project.org/

[44] kernlab. Paquete para métodos kernels en R. Alexandros Karatzoglou

[45] Schölkopf Bernhard Fast protein classification with multiple networks. Bioinformatics,2005, vol21 ,pg ii59-ii65

[46] Scholkopf B, Smola A (2002). Learning with Kernels. MIT Press

[47] Irwin h Segel,Enzyme kinetics, A wiley-Interscience publication pg7 1981

[48] Porter G.J Bartlett, and J. Thornton. The catalytic site atlas, Nucleic Acids Research, 2004, Vol. 32.

[49] Karatzoglou, Alexandros; Meyer David: Support Vector Machines in R ,Report Series / Department of Statistics and Mathematics, Nr. 21, May 2005, GfKl 2006, Berlin, Germany.

[50] Natalia Becker.Elastic SCAD as a novel penalization method for MSV classification tasks in high-dimensional data. BMC Bioinformatics 2011, 12:138.

[51] Hastie T, Tibshirani R, Friedman J: The elements of statistical learning: data mining inference and prediction New York: Springer; 200.

[52] Kuhn, H. W.; Tucker, A. W. (1951). "Nonlinear programming". Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press. pp. 481–492. MR47303