



Facultad de Ingeniería

Escuela de Ingeniería en Bioinformática

Discriminación entre sitios de unión a metales mediante el uso de Máquinas de Vectores de Soporte y la combinación de diversos tipos de información

Alfredo Jesús Pereira Toloza

Profesor Tutor: José Antonio Reyes

Profesor Co-Tutor: Mauricio Arenas

Profesor Informante: Gonzalo Riadi

Memoria para optar al título de Ingeniero en Bioinformática

Talca-Chile

16 de agosto del 2012



UNIVERSIDAD DE
TALCA

Facultad de Ingeniería

Escuela de Ingeniería en Bioinformática

Discriminación entre sitios de unión a metales mediante el uso de Máquinas de Vectores de Soporte y la combinación de diversos tipos de información

Alfredo Jesús Pereira Toloza

José Antonio Reyes : _____

Profesor Tutor

Mauricio Arenas : _____

Profesor Co-Tutor

Gonzalo Riadi : _____

Profesor Informante

Talca-Chile

16 de agosto del 2012

AGRADECIMIENTOS

En primer lugar agradezco a mi familia, por su apoyo y amor durante este proceso, por enseñarme que todo se puede conseguir con esfuerzo y que nunca hay que rendirse cuando las cosas son difíciles.

También agradezco a mis amigos, siempre confiables y alegres, que además han compartido su conocimiento en esta investigación desinteresadamente.

Por último agradezco a mis profesores tutores, por guiar esta investigación, por estar ahí para resolver cualquier duda y por preocuparse siempre que mis resultados sean los mejores.

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS	3
ÍNDICE DE TABLAS.....	5
ÍNDICE DE FIGURAS	7
RESUMEN	8
ABSTRACT	9
INTRODUCCIÓN	10
1.- Metalómica y metaloproteómica	10
1.1.- Metaloproteínas e iones metálicos	10
1.2.- La importancia de Zinc, Magnesio y Calcio	11
1.3.- Sitios de unión a metales.....	12
2.- Predicción de sitios de unión a metales.....	14
2.1.- ¿Por qué predecir sitios de unión a metales?	14
2.2.- Predicción de sitios de unión a metales utilizando <i>Machine Learning</i>	15
2.2.1.- ¿Qué es <i>Machine Learning</i> ?	15
2.2.2.- Proceso de <i>Machine Learning</i>	18
2.2.3.- Métodos de predicción basados en <i>Machine Learning</i>	19
OBJETIVO GENERAL	23
OBJETIVOS ESPECÍFICOS	23
MATERIALES Y MÉTODOS.....	24
1.- Construcción del set de datos no redundante.....	24
2.- Extracción de sitios de unión y estimación de atributos.....	25
2.1.- Estimación de atributos Evolutivos	27
2.1.1.- Puntajes de Matrices de Puntuación de Posición Específica	27
2.2.- Estimación de atributos Geométricos	27
2.2.1.- Distancias	28
2.2.2.- Ángulos.....	28
2.3.- Estimación de atributos Fisicoquímicos	28
2.3.1.- Hidrofobicidades	28
2.3.2.- Composición atómica	28

2.3.3.- Conteo de átomos	29
2.3.4.- Energías	29
2.3.5.- Área accesible al solvente	29
2.3.6.- Composición aminoacídica.....	30
3.- Entrenamiento y validación del modelo basado en SVM	30
4.- Comparación de SVM vs otros modelos supervisados.....	35
RESULTADOS.....	36
1.- Set de datos no redundante.....	36
2.- Atributos	37
2.1.- Atributos Evolutivos	37
2.2.- Atributos Geométricos	39
2.3.- Atributos Fisicoquímicos	41
3.- Desempeño de los modelos de predicción basados en SVM	44
4.- SVM contra otros modelos de aprendizaje supervisado	49
DISCUSIÓN	51
CONCLUSIONES.....	55
REFERENCIAS.....	57
ANEXOS	64

ÍNDICE DE TABLAS

Tabla 1. Grupos funcionales de aminoácidos que interaccionan con metales.....	13
Tabla 2. Cantidad de Cadenas-M, Cadenas-M-NR (no redundantes) y sitios de unión a metales obtenidos para el estudio.....	36
Tabla 3. Valores promedio y desviaciones estándar de atributos evolutivos para cada metal.....	39
Tabla 4. Valores promedio y desviaciones estándar de atributos geométricos basados en distancias para cada metal.....	40
Tabla 5. Valores promedio y desviaciones estándar de atributos geométricos basados en ángulos para cada metal.....	40
Tabla 6. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en hidrofobicidades para cada metal.....	42
Tabla 7. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en composición atómica para cada metal.....	42
Tabla 8. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en conteo de átomos para cada metal.....	43
Tabla 9. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en energías para cada metal.....	43
Tabla 10. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en área accesible al solvente para cada metal.....	43
Tabla 11. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en composición aminoacídica para cada metal.....	44
Tabla 12. Medidas de desempeño para la discriminación entre sitios de unión a metales usando SVM.....	45
Tabla 13. Correlaciones entre atributos Evolutivos basados en Pseudo-PSSM.....	47

Tabla 14. Correlaciones entre atributos Fisicoquímicos basados en Hidrofobicidades.....	47
Tabla 15. Correlaciones entre atributos Fisicoquímicos basados en Energías.....	47
Tabla 16. Atributos que representan ruido para SVM con el aumento de exactitud producido al ser removidos del modelo.....	48
Tabla 17. Medidas de desempeño para SVM con reducción de dimensionalidad.....	48
Tabla 18. Medidas de desempeño por clase para distintos moldes de aprendizaje supervisado.....	50

ÍNDICE DE FIGURAS

Figura 1. Frecuencias de metaloproteínas por tipo ion metálico presentes en PDB.....	12
Figura 2. Crecimiento de la cantidad de proteínas con función desconocida en PDB.....	15
Figura 3. Representación de la forma de operar de SVM.....	18
Figura 4. Representación del efecto de una función <i>kernel</i>	18
Figura 5. Tópicos de bioinformática abordados con <i>Machine Learning</i>	19
Figura 6. Representación de la definición de un sitio de unión a metal.....	26
Figura 7. Distribución de aminoácidos para los sitios de unión a Calcio.....	37
Figura 8. Distribución de aminoácidos para los sitios de unión a Magnesio.....	38
Figura 9. Distribución de aminoácidos para los sitios de unión a Zinc.....	38
Figura 10. Exactitudes obtenidas usando SVM y diversos tipos de información.....	45
Figura 11. Distribución de los coeficientes de correlación entre atributos del modelo.....	46
Figura 12. Exactitudes logradas por Arboles de Decisión (DT), Clasificador Bayesiano Ingenuo (NBC), Regresión Logística (LR) y Máquinas de Vectores de Soporte (SVM).....	49

RESUMEN

Las metaloproteínas son proteínas que requieren la interacción con un metal para desempeñar su función biológica. Aproximadamente un tercio de las proteínas unen al menos un ion metálico en su estructura. Actualmente, en el *Protein Data Bank*, las metaloproteínas más abundantes son aquellas que unen de Zn, Mg o Ca. Los iones metálicos pueden ayudar por ejemplo, a estabilizar la estructura tridimensional de una proteína, inducir cambios conformacionales para regular la función de una proteína, y participar directamente en la actividad catalítica de enzimas. La región de una proteína que une un metal específico es llamada sitio de unión de metal. Conocer estos sitios de unión es fundamental para entender la función de una proteína, pero su identificación implica procedimientos experimentales largos y costosos. Con este fin, diferentes métodos computacionales han sido propuestos recientemente para la predicción de estos sitios de unión, incluyendo diversos enfoques basados en técnicas de *Machine Learning*. La mayoría de los métodos previos han empleado un sólo tipo de información a la vez, ya sea evolutiva, geométrica o fisicoquímica. Además, los trabajos más recientes están mayormente enfocados en predecir sitios de unión para un sólo tipo de metal a la vez, principalmente de Zn, sin la capacidad de discriminar entre sitios de unión a diferentes metales. En el presente estudio, se ha desarrollado un método para predecir y discriminar sitios de unión a Zn, Mg y Ca, utilizando Máquinas de Vectores Soporte y la combinación de información evolutiva, geométrica y fisicoquímica. El método ha sido capaz de distinguir de manera eficiente entre estos sitios de unión, alcanzando una exactitud superior al 90% y una tasa de falsos positivos que no excede del 5%. Además, el estudio proporciona evidencia de que la integración de los diversos tipos de información mejora el desempeño de la discriminación entre diferentes sitios de unión a metales cuando un sólo tipo de información no es suficiente.

ABSTRACT

Metalloproteins are proteins that require interaction with a metal to perform its biological function. Approximately one third of the proteins bind at least one metal ion in its structure. Currently at the *Protein Data Bank*, the most abundant metalloproteins are those binding Zn, Mg or Ca. Metal ions may help for instance, to stabilize the three dimensional structure of a protein, to induce conformational changes for protein function regulation, and directly participating in the enzyme catalytic activity. The region of a protein which binds a specific metal is called metal binding site. Knowing these binding sites is critical to understand the function of a protein, but identifying them involves long and costly experimental procedures. In order to do this, different computational methods have been recently proposed for the prediction of these binding sites, including various approaches focused on Machine Learning techniques. Most of the previous methods have employed only one type of information at the time, whether evolutionary, geometrical or physicochemical. In addition, the most recent attempts are mostly concerned to predict binding sites for only one type of metal at the time, mainly Zn, without the ability to discriminate between different metal binding sites. In the present study, it has been developed a method for prediction and discrimination of binding sites to Zn, Mg and Ca, using Support Vector Machines and the combination of evolutionary, geometric and physicochemical information. The method has been able to distinguish efficiently between these binding sites, reaching an accuracy over 90% and a false positive rate that do not exceed 5%. In addition, the study provides evidence that the integration of various types of information improves the performance of the discrimination between different metal binding sites when a single type of information is not enough.

INTRODUCCIÓN

1.- Metalómica y metaloproteómica

La metalómica y metaloproteómica son campos relativamente nuevos que abordan la función, captación, transporte y almacenamiento de los metales esenciales para la vida, sin embargo la genómica y la proteómica, áreas desarrolladas desde hace varios años han permitido la construcción de una gran cantidad de datos que pueden ser utilizados en estudios de estas dos nuevas disciplinas (Shi y Chance, 2008). Metalómica se define como el análisis de la totalidad de los metales y metaloides dentro de una célula o un tipo de tejido, mientras que la metaloproteómica se enfoca en explorar la función de las metaloproteínas y su asociación con los metales (Szpunar, 2005) (Bertini y Cavallaro, 2008).

1.1.- Metaloproteínas e iones metálicos

Las metaloproteínas son proteínas que necesitan de la interacción con un cofactor metálico para desempeñar su función biológica (Finkelstein, 2009). Las metaloproteínas participan en diversos procesos bioquímicos como estabilidad estructural (Cox y McLendon, 2000), regulación de la expresión de genes (Bouton y Pevsner, 2000), procesamiento de ADN (Feng y col., 2004), procesos de señalización (Carafoli, 2002), control del metabolismo (Harris, 2000), homeostasis de metales (Cobbett y Goldsbrough, 2002) y reconocimiento de anticuerpos (Zhou y col., 2005), además de contribuir en eventos tales como, la respiración celular, movimiento muscular y defensa antioxidante (Lieu y col., 2001).

Se ha descrito que un tercio de las proteínas unen uno o más metales en su estructura (Holm y col., 1996), actualmente 23.476 de las 76.220 proteínas depositadas en *Protein Data Bank* (PDB) (Berman y col., 2000) contienen algún tipo

de ion metálico en su estructura (datos con fecha 16 de junio del 2012). Los iones metálicos pueden ayudar por ejemplo, a estabilizar la estructura tridimensional de una proteína, inducir cambios conformacionales para regular la función de una proteína, y participar directamente en la actividad catalítica de enzimas (Degtyarenko, 2000).

1.2.- La importancia de Zinc, Magnesio y Calcio

De la gama de iones metálicos que interaccionan con proteínas, tres son foco de este estudio debido a su importancia biológica, estos iones son el Zinc, el Magnesio y el Calcio. En primer lugar, las metaloproteínas con estos tres tipos de iones metálicos son las más abundantes en PDB (Ver figura 1), y sólo estos tres tipos de proteínas representan actualmente un cuarto de las estructuras de proteínas disponibles en esta base de datos. Por otro lado, estos metales juegan roles esenciales del punto de vista bioquímico. El Zinc es un cofactor esencial para factores de transcripción, el Magnesio es de vital importancia para el proceso de replicación del ADN, ya que activa enzimas involucradas en tal proceso, como la polimerasa y el Calcio cumple importantes roles dentro de la célula como fuera de ella, debido a su función como mensajero secundario, destacando la regulación de la contracción muscular, la glucólisis y la gluconeogénesis (Crichton, 2008).

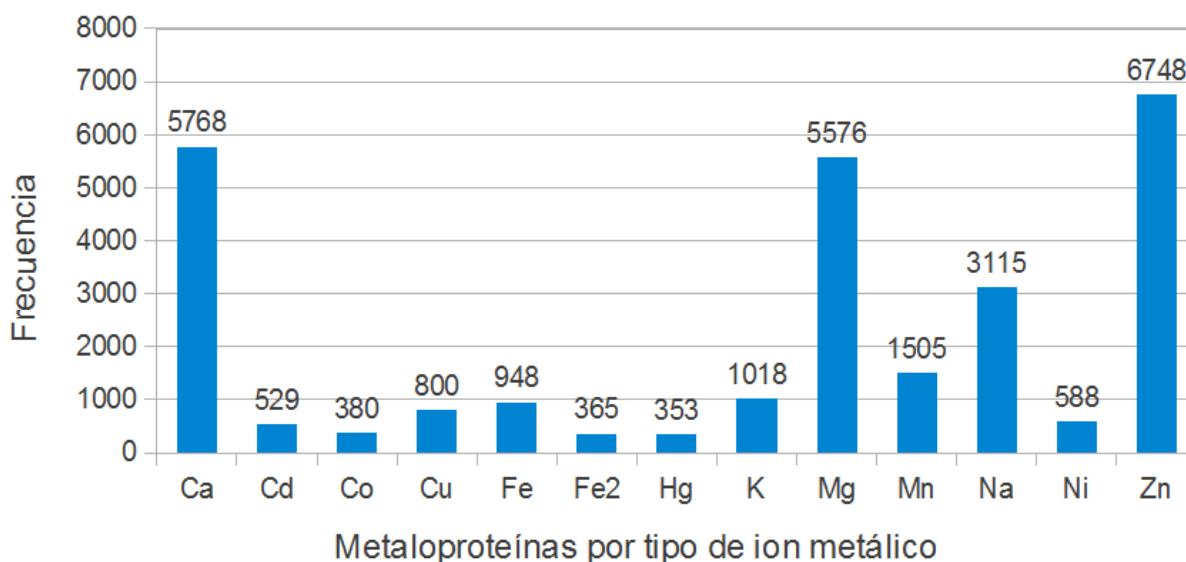


Figura 1. Frecuencias de metaloproteínas por tipo ion metálico presentes en PDB. Datos obtenidos desde www.pdb.org el 16 de junio del 2012.

1.3.- Sitios de unión a metales

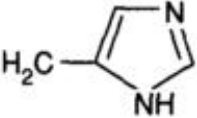
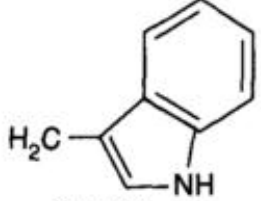
Un sitio de unión a metal es la región de una proteína (conjunto de aminoácidos) donde se une un metal específico, y están usualmente localizados en cavidades o hendiduras de la proteína (Dudev y Lim, 2003). Estos sitios de unión pueden ser clasificados en 5 tipos básicos según la función que cumplen (Holm y col., 1996):

- 1) Estructural: configura la estructura terciaria o cuaternaria de la proteína
- 2) Almacenamiento: capta, une y entrega metales en forma soluble
- 3) Transferencia de electrones: capta y sede electrones
- 4) Unión de oxígeno: coordina y descoordina la unión metal-O₂
- 5) Catalítica: une sustrato y activa la catálisis

Los grupos funcionales de los aminoácidos que con mayor frecuencia se unen a iones metálicos son los grupos carboxílico, imidazol, indol, tiol, tioéter, hidroxilo y

posiblemente los grupos amida (Anfinsen y col., 1991) (Ver tabla 1). Las cadenas laterales de 13 de los 20 tipos de aminoácidos son potenciales grupos de unión para iones metálicos.

Tabla 1. Grupos funcionales de aminoácidos que interaccionan con metales. Tomada de (Anfinsen y col., 1991)

Group	Abbreviation	Formula	Metal-binding atom
Carboxyl	Asp	CH_2COO^-	O
	Glu	$\text{CH}_2\text{CH}_2\text{COO}^-$	O
Imidazole	His		N
Indole	Trp		N
Thiol	Cys	CH_2SH	S
Thioether	Met	$\text{CH}_2\text{CH}_2\text{SCH}_3$	S
Hydroxyl	Ser	CH_2OH	O
	Thr	$\text{CH}(\text{CH}_3)\text{OH}$	O
	Tyr	$\text{CH}_2-\text{C}_6\text{H}_4-\text{OH}$	O
Amide	Asn	CH_2CONH_2	O
	Glu	$\text{CH}_2\text{CH}_2\text{CONH}_2$	O
Amino ^a	Lys	$(\text{CH}_2)_4\text{NH}_3^+$	N
	Arg	$(\text{CH}_2)_3\text{NH}-\text{C} \begin{matrix} \nearrow \text{NH}_2^+ \\ \searrow \text{NH}_2 \end{matrix}$	N
Main-chain carbonyl		$\text{C}=\text{O}$	O
Main-chain amino ^a		$\text{N}-\text{H}$	N

^a Questionable.

2.- Predicción de sitios de unión a metales

2.1.- ¿Por qué predecir sitios de unión a metales?

Comprender los mecanismos moleculares que gobiernan el funcionamiento de una proteína es una etapa clave para entender los complejos procesos que envuelven los organismos vivos (Cilia y Passerini, 2010). La mayoría de estos procesos biológicos son determinados por la interacción de cierta proteína con otra molécula o cofactor (Capra y col., 2009). La función bioquímica de una proteína no sólo depende de su conformación estructural final, sino que además es dictado por la ubicación de residuos funcionales que interactúan con otra molécula, como por ejemplo los iones metálicos (Goyal y Mande, 2008).

Con la predicción de estos residuos funcionales, o en este estudio sitios de unión a metales, se puede obtener conocimiento importante para comprender la función que desempeña una proteína no caracterizada. Desde el año 2000 la cantidad de estructuras de proteínas con función desconocida ha aumentado significativamente (Ver figura 2), y actualmente en PDB existen 3025 proteínas con función desconocida. Además, determinar estos sitios de unión puede ser muy útil en el diseño de fármacos, permitiendo diseñar inhibidores de un sitio funcional. Entender la interacción de iones metálicos con proteínas podría guiar el desarrollo de nuevos fármacos para ser usados en el tratamiento de enfermedades tales como el Parkinson y el Alzheimer (Barnham y Bush, 2008).

Experimentalmente, las metaloproteínas son identificadas y/o caracterizadas usando espectroscopia de absorción (Reed y Poyner, 2000), electroforesis en gel (Binet y col, 2003), cromatografía (Papoyan y Kochian, 2004), espectroscopia de masas (Binet y col., 2003) y resonancia magnética nuclear (Jensen y col., 2005). La mayoría de estos métodos requieren etapas complejas y equipamiento especializado, haciendo de ellos procesos largos y costosos. Por lo tanto, existe una necesidad de explorar otros métodos, incluyendo enfoques computacionales, para

facilitar la identificación de metaloproteínas y de los sitios de unión donde ocurre la interacción con el metal.

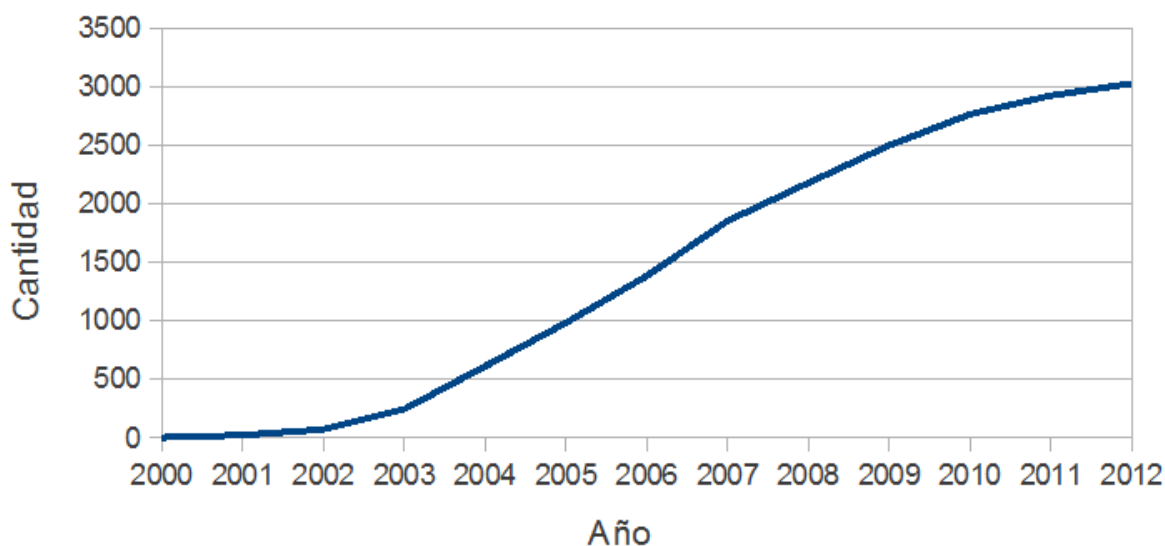


Figura 2. Crecimiento de la cantidad de proteínas con función desconocida en PDB. Datos obtenidos desde www.pdb.org el 22 de julio del 2012.

Actualmente, gracias a la gran cantidad de datos biológicos, ya sea secuencias de aminoácidos o estructuras tridimensionales de proteínas disponibles en PDB, es factible la extracción de información útil para la generación de sistemas inteligentes capaces de predecir sitios de unión metales en proteínas.

2.2.- Predicción de sitios de unión a metales utilizando *Machine Learning*

2.2.1.- ¿Qué es *Machine Learning*?

Según (Mitchell, 1997), *Machine Learning* es un proceso en el cual un computador aprende de una experiencia E con respecto a una tarea de aprendizaje T, la cual es medida por una performance P y se busca que la performance (de ahora en adelante desempeño) P en la tarea T, mejore con la experiencia E. Otras fuentes definen *Machine Learning* como una área de la inteligencia artificial, la cual se refiere al desarrollo de algoritmos y técnicas que permitan a los computadores aprender

(Zhang y Rajapakse, 2008), en otras palabras consiste en la programación de algoritmos para optimizar un criterio de aprendizaje, mediante el uso de datos (Larranaga, 2006).

Es posible reconocer dos principales categorías de modelos de *Machine Learning*:

- 1) Modelos de aprendizaje supervisado: en este caso el objetivo es predecir la clase de un dato en base a mediciones que este pueda tener, las que son recibidas como entrada para generar el modelo. Aquí la clase es conocida (Hastie y col, 2009). Las tareas principales son la clasificación y la regresión.
- 2) Modelos de aprendizaje no-supervisado: en este caso no hay una clase asociada a los datos y el objetivo es describir una cantidad de patrones entre los datos de entrada conocidos (Hastie y col., 2009). Las tareas principales son las reglas de asociación y el *clustering*.

El presente estudio se ha enfocado en una tarea de clasificación supervisada, donde la idea es discriminar entre de sitios de unión a Ca, Mg y Zn. Algunos de los algoritmos de aprendizaje supervisados más comunes son:

- 1) Redes neuronales artificiales (ANN, del inglés *Artificial Neural Networks*): Es un grupo interconectado de nodos que usa un modelo computacional para procesar información, cambia su estructura en base a información externa o interna que fluye a través de la red. Una red neuronal artificial puede ser usada para modelar una relación compleja entre entradas y salidas, y encontrar patrones en los datos (Mitchell, 1997).
- 2) Árboles de decisión (DT, del inglés *Decision Tree*): Es una técnica de aprendizaje supervisado que usa aproximación de funciones discretas para estimar y clasificar los ejemplos. Cada hoja es una clasificación y cada rama es un conjunto de atributos que lleva a esa clasificación. Los árboles de decisión pueden entregar una estimación de probabilidad de ocurrencia de un

caso particular (Mitchell, 1997).

- 3) *Random forest* (RF): Es un clasificador que consiste de la combinación de varios árboles de decisión. *Random forest* entrega como salida la clase que es la moda de las clases entregadas por los árboles de decisión individuales (Mitchell, 1997).
- 4) Clasificador bayesiano ingenuo (NBC, del inglés *Naive Bayes Classifier*): Es un clasificador bayesiano que clasifica en base a las reglas bayesianas de probabilidad condicional. Utiliza todos los atributos, permitiéndoles hacer contribuciones a las decisiones como si fueran todos igualmente importantes e independientes uno de otro (Mitchell, 1997).
- 5) Regresión logística (LR, del inglés *Logistic Regression*): Es un modelo usado para estimar la probabilidad de ocurrencia de un evento mediante datos ajustados a una curva logística. Esta hace uso de variables numéricas o categóricas. Por ejemplo, la probabilidad que una persona tenga un ataque al corazón en un periodo de tiempo específico debería ser predicho desde la edad, el sexo y la masa corporal de la persona (Mitchell, 1997).
- 6) Máquinas de vectores de soportes (SVM, del inglés *Support Vector Machines*) (Vapnik, 1998): Es un método de aprendizaje supervisado, que ha aumentado su uso y popularidad en los últimos años, desplazando a otros métodos como redes neuronales artificiales, debido a la exactitud de sus predicciones (Bishop, 2007). SVM consiste en buscar un hiper-plano (recta, en el caso más simple) que separe las clases (Ver figura 3), maximizando el margen M entre los datos de cada clase. Pero la mayoría de las veces el problema de clasificación no se encuentra de forma lineal, es por esto que es necesario aplicar una función *kernel* (K), que transforme el espacio del vector de atributos en otro, de tal forma que el problema queda representado de manera lineal (Ver figura 4).

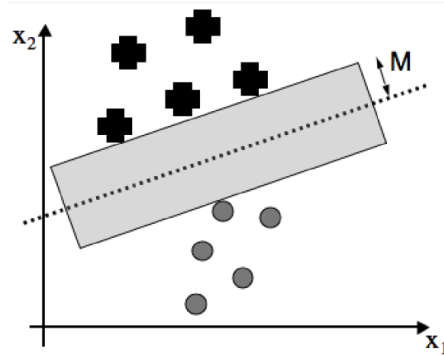


Figura 3. Representación de la forma de operar de SVM. Tomada de (Marsland, 2009).

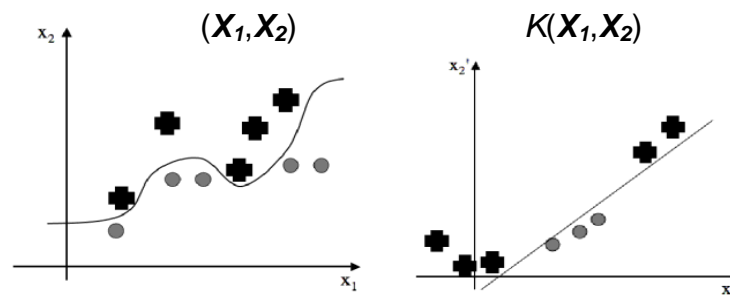


Figura 4. Representación del efecto de una función *kernel*. Adaptada de (Marsland, 2009).

2.2.2.- Proceso de *Machine Learning*

En un problema típico, el término *Learning* se refiere a ejecutar un algoritmo para inducir un modelo mediante el uso de datos de entrenamiento. *Machine Learning* usa teorías estadísticas al momento de construir el modelo computacional donde posteriormente el objetivo es generar inferencias a partir de la muestra de datos, y estas inferencias pasan a ser llamadas conocimiento. Este proceso de transformación de datos en conocimiento se lleva a cabo en varios pasos. En el primer paso es necesario juntar e integrar las diferentes fuentes de datos en un sólo formato. Luego, el segundo paso consiste en seleccionar, limpiar y transformar los datos. Aquí se incluye la eliminación de datos incorrectos y redundantes. En el tercer paso los objetivos del estudio se toman en cuenta, para elegir el tipo de análisis más apropiado, en donde se debe optar por un modelo supervisado o no-supervisado.

Finalmente, el modelo es generado y entonces la información obtenida debería ser evaluada e interpretada (Larranaga, 2006). Métodos de *Machine Learning* son utilizados en distintas áreas de la bioinformática (Ver Figura 5).

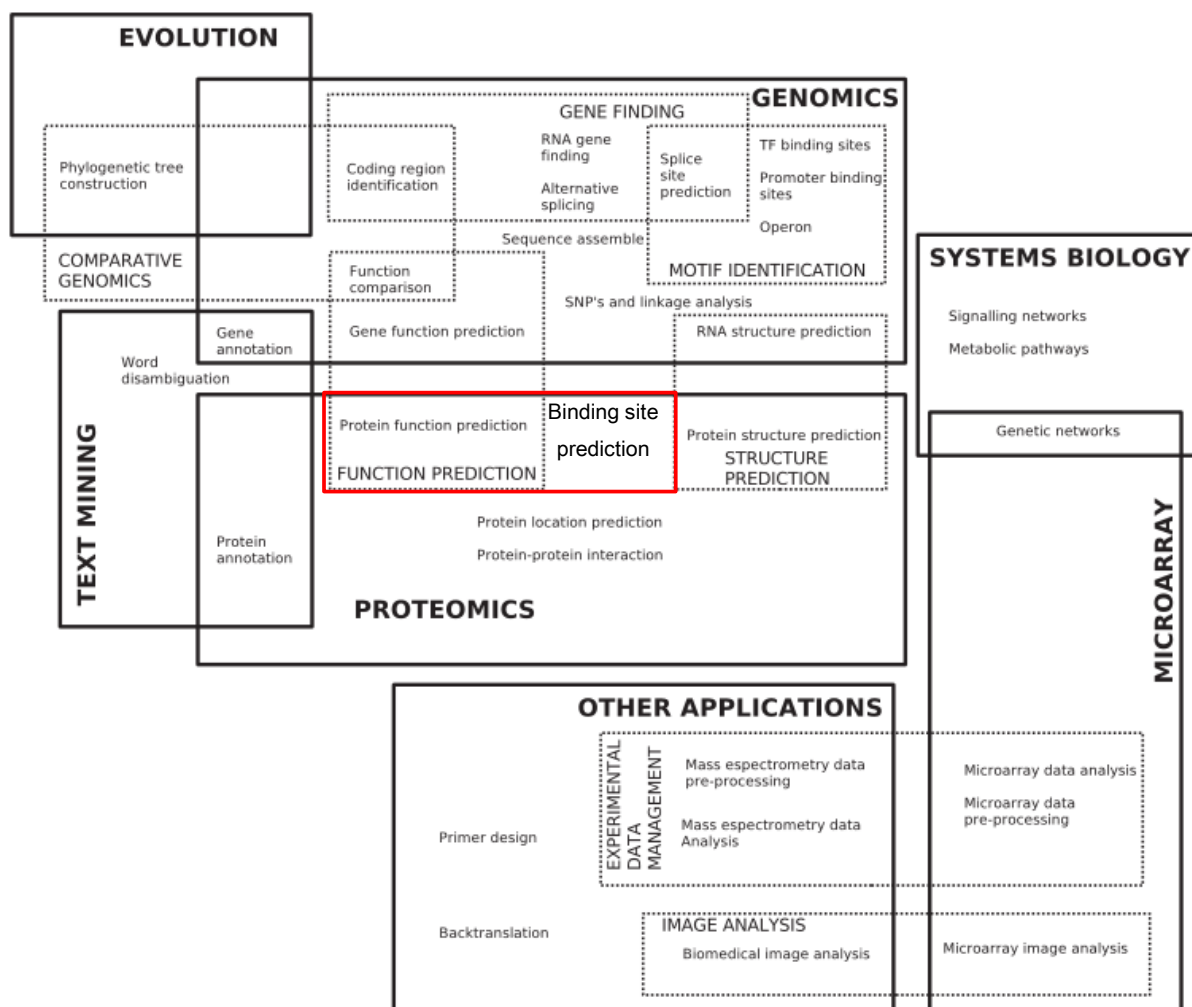


Figura 5. Tópicos de bioinformática abordados con *Machine Learning*. En rojo la área abordada en este estudio. Adaptada de (Larranaga, 2006)

2.2.3.- Métodos de predicción basados en *Machine Learning*

Los sitios de unión a un determinado metal disponibles en las proteínas de PDB son caracterizados por un conjunto de atributos asociados a alguno de los siguientes tipos de información:

- 1) Información evolutiva (Passerini y col., 2006) (Shu y col., 2008): referida a secuencias de aminoácidos conservadas.
- 2) Información geométrica (Sodhi y col., 2004): principalmente medidas de distancia y ángulos.
- 3) Información fisicoquímica (Lin CT. y col., 2005) (Ebert y Altman, 2008): generalmente energías, hidrofobicidades, cargas atómicas, entre otras.

Los métodos de predicción basados *Machine Learning* utilizan estos sitios de unión caracterizados para entrenar y validar determinados modelos de aprendizaje supervisado, tales como ANN, RF, NBC y SVM. La mayoría de estos métodos miden el desempeño de sus modelos con tasas de verdaderos positivos (TP, del inglés *True Positives*), la cual representa la cantidad de sitios bien clasificados. Otros construyen curvas ROC (del inglés *Receiver Operating Characteristic*), las cuales son una representación gráfica de los sitios bien clasificados frente a los mal clasificados. De estas curvas se puede obtener, como medida de desempeño, el área bajo la curva (AUC del inglés *Area Under Curve*).

Diversos métodos han sido desarrollados basándose en información fisicoquímica. (Lin CT. y col., 2005) generaron modelos basados en ANN para predecir de manera independiente sitios de unión Calcio, Potasio, Magnesio y Sodio. Los autores reportan una tasa de TP de 90%. Sin embargo, las tasas varían de 0% a 90%, dependiendo del metal y de los atributos utilizados. Posteriormente (Lin HH. y col., 2006) introdujeron SVM para predecir de manera independiente sitios de unión a Calcio, Cobalto, Cobre, Hierro, Magnesio, Manganeso, Níquel, Potasio, Sodio y Zinc. El desempeño logrado por este método alcanzó una tasa de 78% TP. (Ebert y Altman, 2008) desarrollaron FEATURE, basado en un clasificador bayesiano, para predecir sólo sitios de unión a Zinc. Este método alcanza una tasa de TP del 76%. Además en este estudio propusieron un modelo de discriminación entre sitios de unión Zinc y Calcio. El método no discrimina eficientemente entre estos dos metales,

ya que el resultado de esta clasificación no supera una tasa de 71% de TP.

La información evolutiva también ha sido utilizada en varios estudios. (Passerini y col., 2006) implementaron un método de dos etapas, la primera basada en SVM, y la segunda en ANN. Los modelos generados permitieron predecir de manera independiente sitios de unión a Zinc, Cobre, Cadmio, Hierro, Níquel, y además de complejos Hierro/azufre y grupos Hemo. El método fue capaz de predecir con una tasa de 63% TP. Posteriormente (Passerini y col., 2007) desarrollaron un método para predecir sólo sitios de unión a Zinc, descartando incluir una segunda etapa basada ANN, ya que la mejora no era estadísticamente significativa. El desempeño, medido con el AUC, fue de 0,9. (Shu y col., 2008) también utilizaron SVM para predecir sitios de unión a Zinc. Los resultados muestran una tasa de TP de 70%. (Brylinski y Skolnick, 2011), basándose en SVM, desarrollaron FINDSITE-Metal para predecir de manera independiente sitios de unión a Hierro, Cobre, Zinc, Calcio y Magnesio. Este método alcanzo tasas de TP de sólo 50%.

Algunos métodos han combinado diversos tipos de información para predecir sitios de unión a metales. (Sodhi y col., 2004) utilizaron ANN para predecir de manera independiente sitios a Calcio, Cobre, Hierro, Magnesio, Manganeso y Zinc, mediante la combinación de atributos evolutivos, geométricos y fisicoquímicos. El método logró tasas de TP cercanas al 85%. Sin embargo, en este caso es necesario considerar que, el set de datos de entrenamiento era reducido. (Bordner, 2008) desarrolló SitePredict, un método basado en RF. En este estudio fueron generados modelos para predecir de manera independiente sitios de unión a Calcio, Cobre, Hierro, Magnesio, Manganeso y Zinc. El método combinando atributos evolutivos y fisicoquímicos, logro un desempeño de 0,8 AUC. En este trabajo también se intentó discriminar entre sitios de unión a diferentes metales. Se generó en particular un modelo de discriminación Ca vs Mg. El desempeño logrado no supero el 0,6 AUC, y el autor señala que su método no es capaz de discriminar entre estos sitios de unión a metales.

De los estudios anteriormente expuestos se puede inferir que la mayoría están basados en información evolutiva o fisicoquímica dejando de lado los atributos geométricos. Además, la combinación de distintos tipos de información es poco frecuente. Los métodos recientes se enfocan en un sólo tipo de metal, principalmente en Zinc, y dejan de abordar otros metales de importancia biológica como por ejemplo el Magnesio y el Calcio. Además, estos métodos generan modelos para predecir de sitios de unión a diferentes metales, pero sólo de manera independiente. Sin embargo, dos estudios (Bordner, 2008) y (Ebert y Altman, 2008) intentaron discriminar entre sitios de unión a metales diferentes, pero obteniendo un bajo desempeño que no superaba el 0,6 AUC y el 71% de tasas de TP.

Finalmente, se puede decir que hasta la fecha no existen métodos eficientes basados en *Machine Learning* que permitan discriminar entre sitios de unión a diferentes metales, integrando información evolutiva, geométrica y fisicoquímica, y que al mismo tiempo identifiquen qué tipo de información genera predicciones más acertadas para cada sitio de unión a metal. Entonces, en el presente trabajo se propone como desafío construir un modelo de predicción basado en *Machine Learning* para discriminar entre sitios de unión a Zn, Mg y Ca, superando el desempeño de los métodos reportados hasta la fecha.

OBJETIVO GENERAL

Desarrollar un modelo de predicción basado en Maquinas de Vectores de Soporte que permita predecir y discriminar sitios de unión específicos a Zn, Mg y Ca en metaloproteínas.

OBJETIVOS ESPECÍFICOS

- 1) Construir un set de datos no redundante de estructuras de proteínas.
- 2) Extraer los sitios de unión y estimar los atributos basados en información evolutiva, geométrica y fisicoquímica.
- 3) Entrenar y validar el modelo basado en Máquinas de Vectores de Soporte y encontrar parámetros óptimos.
- 4) Evaluar el desempeño del modelo basado en Maquinas de Vectores de Soporte y compararlo con otros modelos basados en Arboles de Decisión, Clasificador Bayesiano Ingenuo y Regresión Logística.

MATERIALES Y MÉTODOS

1.- Construcción del set de datos no redundante

La construcción del set de datos no redundante es el primer paso en cualquier proceso *Machine Learning*, este consiste en seleccionar, limpiar y transformar los datos de acuerdo a ciertos criterios guiados por la problemática abordada y los objetivos planteados (Larranaga, 2006). Lo primordial en esta etapa es eliminar la mayor cantidad de datos redundantes en el set de datos, para que en etapas posteriores las predicciones no sean sesgadas.

En primer lugar, para la construcción del set de datos no redundantes de estructuras de proteínas, se descargaron desde PDB las estructuras que contuvieran Ca, Mg o Zn, usando el parámetro *Chemical ID* de la búsqueda avanzada. Las proteínas depositadas en PDB pueden estar constituidas por dos o más cadenas, y no necesariamente todas las subunidades tienen sitios de unión a metales. Considerando esto se identificaron sólo aquellas cadenas (desde ahora en adelante Cadenas-M) que unan al menos uno de los metales antes mencionados. Para ello se programó un script en el lenguaje de programación *Perl* (anexo 1.1).

Posteriormente, se eliminaron todas aquellas Cadenas-M que podían ocasionar ruido o sesgo en las predicciones. Para ello, se utilizó PISCES (Wang y Dunbrack, 2003), el cual es un servidor *Web* que permite eliminar estructuras de proteínas redundantes mediante el agrupamiento de las secuencias aminoacídicas en base a su porcentaje de identidad. De acuerdo a lo mencionado por (Chothia y Lesk, 1986), las estructuras tridimensionales de las proteínas son mucho más conservadas que las secuencias aminoacídicas, en donde un porcentaje de identidad de secuencia sobre 25% conlleva estructuras altamente similares. Por otro lado, estructuras de alta resolución son necesarias para identificar sitios de unión a metales (Wei y col., 1999). Considerando lo anterior los parámetros utilizados en

PISCES fueron:

- ✓ Porcentaje de identidad para agrupamiento de secuencias: 25%
- ✓ Intervalo de Resolución: 0 ~ 2 Å

Los siguientes parámetros no fueron modificados, ya que son recomendados por PISCES para obtener estructuras de mayor calidad.

- ✓ Intervalo de R-factor: 0 ~ 0,3
- ✓ Largo de secuencias: 60 ~ 600 aminoácidos
- ✓ Excluir estructuras que no fueron cristalizadas por difracción de rayos X
- ✓ Excluir estructuras constituidas sólo de C α

2.- Extracción de sitios de unión y estimación de atributos

La definición y extracción de los sitios de unión a metales en las metaloproteínas se realizó mediante el enfoque de (Bagley y Altman, 1995), utilizado ya en estudios anteriores (Ebert y Altman, 2008) (Nassif y col., 2009).

Los sitios de unión a metales se definieron como una esfera de radio 7 Å, centrada en el ion metálico (Ver figura 6). La utilización de este *cutoff* permite considerar interacciones de tipo no-enlazantes de largo alcance que pueden ser significativas para los iones metálicos (Kumar y Nussinov, 2002). Para extraer estos sitios de unión se construyó un script en el lenguaje de programación *Tcl* (anexo 1.2) para ser ejecutado en el software *Visual Molecular Dynamics* (VMD) (Humphrey y col., 1996). Este permitió seleccionar y extraer todos los sitios de unión a metales presentes en las Cadenas-M no redundantes seleccionadas en el paso anterior. Se consideraron además 3 condiciones que debían cumplir los sitios de unión:

- 1) El sitio de unión al metal debía contener al menos 4 aminoácidos. Parámetro elegido considerando el número de coordinación menor entre los 3 metales.
- 2) El sitio de unión al metal debía contener al menos 1 aminoácido a menos de 3 Å. Parámetro usado por estudios anteriores (Dokmanić y col., 2008).
- 3) No se consideraron sitios de unión sobrelapados. Parámetro elegido para descartar sitios redundantes.

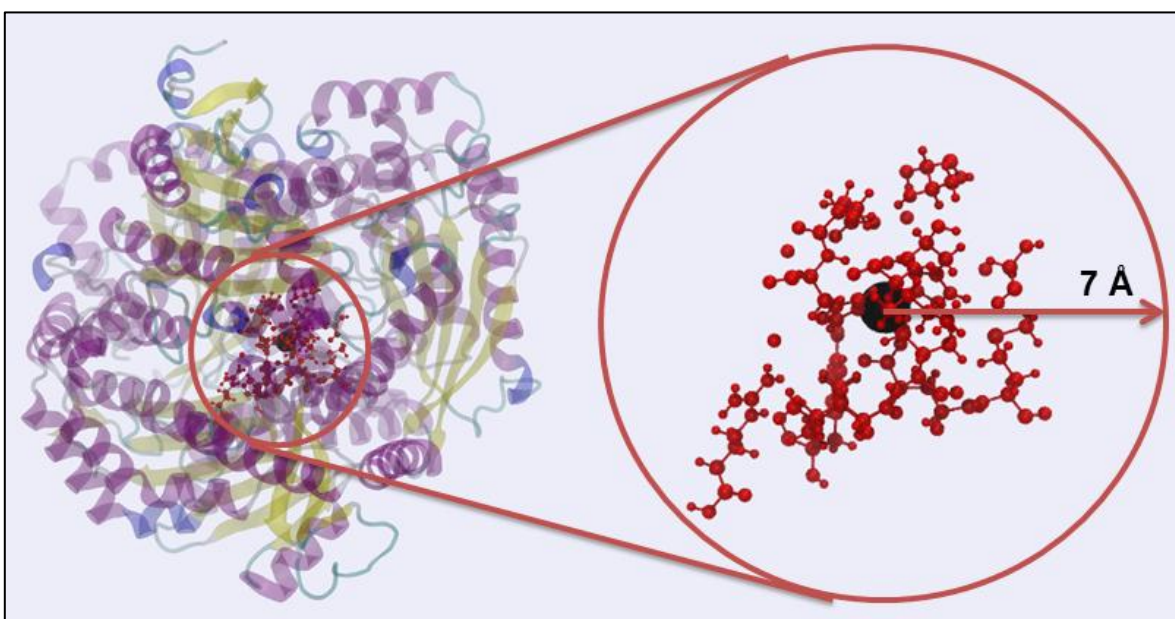


Figura 6. Representación de la definición de un sitio de unión a metal.

Luego, mediante un script en el lenguaje de programación *Perl* (anexo 1.3) se descartaron aquellos sitios de unión que presentaban residuos con dos o más conformaciones, ya que atributos como distancias, ángulos y conteo de átomos calculados sobre estos sitios serían incorrectos.

La estimación y selección de atributos es la etapa más compleja y que lleva más tiempo realizar (Larranaga, 2006). Para el presente estudio se integraron atributos de tipos evolutivos, geométricos y fisicoquímicos.

2.1.- Estimación de Atributos Evolutivos

Los atributos evolutivos fueron determinados considerando sólo los 4 aminoácidos más cercanos al metal, ya que todos los sitios de unión utilizados en este estudio contienen al menos 4 aminoácidos. Es importante destacar que, estos 4 aminoácidos no son necesariamente aquellos con los que el metal está coordinado. La identificación de los 4 aminoácidos más cercanos al metal se realizó mediante un script en lenguaje de programación *Perl* (anexo 2.1).

2.1.1.- Puntajes de Pseudo-PSSM

Puntajes de Matrices de Puntuación de Posición Específica (PSSM, del inglés *Position-Specific Scoring Matrix*) ya han sido utilizados en estudios anteriores (Sodhi y col., 2004) (Shu y col., 2008). Para el presente estudio se utilizaron Pseudo-PSSM, denominadas de esta forma ya que el orden de la secuencia aminoacídica del sitio de unión es dictado, en este caso, por la cercanía al metal. Para el cálculo de los puntajes se utilizó la suite de programas EMBOSS (Rice y col., 2000). En primer lugar *Prophecy* fue utilizado para crear las 3 PSSM (una por cada tipo de sitio de unión a metal), recibiendo como entrada los 3 set de secuencias alineadas y entregando como salida las matrices en formato de texto plano. Luego con *Profit* se escanearon las secuencias de cada sitio con cada una de las 3 matrices generadas anteriormente, asignando a cada sitio de un unión 3 puntajes.

2.2.- Estimación de Atributos Geométricos

Los atributos geométricos fueron calculados para los 4 aminoácidos más cercanos al metal. Fue necesario construir un script en *Perl* (anexo 2.2) el cual determinara las coordenadas X, Y, Z del ion metálico (M), del átomo de la cadena lateral (R) más cercano al metal para cada uno de los 4 aminoácidos y del carbono alfa (C α) para cada uno de los 4 aminoácidos, luego estas coordenadas fueron escritas en un archivo CSV para su posterior uso. Atributos geométricos tales como distancias y ángulos ya han sido utilizados en estudios anteriores (Goyal y Mande, 2008).

2.2.1.- Distancias

Para el cálculo de distancias fue construido un script en *Perl* (anexo 2.3), con el cual se leyeron las coordenadas desde el archivo CSV y mediante el uso del teorema de Pitágoras se obtuvieron las distancias en Å para pares de átomos determinados.

2.2.2.- Ángulos

Para el cálculo de los ángulos, fue utilizado el script en *Perl* del paso anterior, pero además fue implementado en *python* (anexo 2.4) la fórmula del producto escalar, la cual permitió obtener el ángulo entre 2 vectores conociendo las coordenadas de 3 átomos.

2.3.- Estimación de Atributos Fisicoquímicos

2.3.1.- Hidrofobicidades

Las hidrofobicidades de residuos ya han sido utilizadas en estudios anteriores (Ebert y Altman, 2008) (Lin CT. y col., 2005) para caracterizar sitios de unión a metales. Estas se estimaron considerando sólo los 4 aminoácidos más cercanos al metal. Para la asignación de hidrofobicidades, fue construido un script en *Perl* (anexo 2.5), con el cual se leyó la secuencia en formato fasta de los 4 residuos más cercanos (generada para los atributos evolutivos) y entregaba las medidas de hidrofobicidad para cada aminoácido. Fueron utilizadas tres escalas de hidrofobicidad (Kyte y Doolittle, 1982) (Wimley y White, 1996) (Hessa y col., 2005), las cuales serán revisadas en el anexo 4.

2.3.2.- Composición Atómica

La composición atómica por capas fue anteriormente utilizada por (Ebert y Altman, 2008). Aquí el sitio de unión fue dividido en capas de 1 Å cada una, y la

composición atómica fue calculada para cada una de las capas, a excepción de la primera más cercana al metal en la cual no fueron encontrados átomos. Para esto se programó un script en *Tcl* (anexo 2.6) para ser ejecutado en VMD. Se obtuvieron luego los porcentajes de Carbono, Oxígeno, Nitrógeno y Azufre.

2.3.3.- Conteo de Átomos

El conteo de átomos también fue cálculo por capas. La primera capa también fue descartada. Para calcular la suma de todos los átomos presentes en cada capa se utilizó el script del paso anterior.

2.3.4.- Energías

Diferentes tipos de energías fueron calculadas usando el campo de fuerza FoldX (Guerois y col., 2002), el cual ya fue utilizado anteriormente por otros métodos de predicción de sitios de unión (Schymkowitz y col., 2005). Las energías fueron calculadas considerando todos los aminoácidos del sitio de unión. FoldX es un campo de fuerza empírico que permite el cálculo de energías libres de proteínas. Cada una de las energías fue obtenida en Kcal/mol.

2.3.5.- Área Accesible al Solvente

El área accesible al solvente ha sido utilizado como atributo en modelos de predicción de sitios de unión a metales (Sodhi y col., 2004) (Ebert y Altman, 2008) (Lin CT. y col., 2005). Esta fue calculada para el sitio de unión completo. Para calcular el área accesible al solvente de los sitios de unión se utilizó la función *measure sasa* de VMD en un script en *Tcl* (anexo 2.7).

2.3.6.- Composición Aminoacídica

La composición aminoacídica de los sitios de unión fue estimada con un script en *Tcl* (anexo 2.7) para ser ejecutado en VMD. Los aminoácidos fueron agrupados de acuerdo al tipo de carga en sus cadenas laterales a pH neutro.

Un resumen de estos atributos puede ser revisado en el anexo 6. Para otorgar persistencia a los datos estos fueron almacenados en una base de datos administrada con MYSQL. Se crearon tres tablas, una para cada tipo de información, además de una cuarta tabla con todos los atributos, en la cual los sitios de unión fueron ordenados de manera aleatoria, pero asegurando el balance de clases necesario para etapas posteriores.

3.- Entrenamiento y validación del modelo basado en SVM

El algoritmo de aprendizaje supervisado utilizado fue SVM (Vapnik, 1998), el cual fue implementado con las librerías de LIBSVM (Chang y Lin, 2011), en un entorno Matlab. A continuación se entrega una breve descripción de los fundamentos matemáticos de SVM.

Considerando un set de entrenamiento dado por:

$$D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}, i = 1, \dots, m \quad (1)$$

Donde \mathbf{x}_i es un vector de dimensionalidad n , e y_i es -1 o 1 indicando la clase de cada ejemplo \mathbf{x}_i , SVM soluciona el siguiente problema de optimización cuadrática (forma dual):

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^n} \quad & \frac{1}{2} \alpha^T \Omega \alpha - \mathbf{e}^T \alpha \\ \text{Sujeto a} \quad & \mathbf{y}^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \\ & i = 1, \dots, m \end{aligned} \tag{2}$$

Donde \mathbf{e} es un vector de dimensionalidad n de unos, α es un vector de dimensionalidad m de variables duales, C es una cota superior para valores α_i , Ω es una matriz positiva semi-definida de m por m ,

$$\Omega_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

Y $K(\mathbf{x}_i, \mathbf{x}_j)$ es una función *kernel* usada para la creación de un clasificador no lineal. Se consideró sólo la función de base radial la cual es recomendada en la literatura (Hsu y col., 2003).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0 \tag{4}$$

Usualmente, una función de respuesta es usada para producir una función de decisión desde la salida sin umbrales de SVM.

$$f(\mathbf{x}_j) = \text{sgn} \left(\sum_{i=1}^m y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) + b \right) \tag{5}$$

Donde la función f representa la salida con umbrales de SVM, b es un término de sesgo y sgn es una función de respuesta para producir una salida con umbrales

de SVM desde una sin umbrales. Cada ejemplo nuevo es asignado a la clase correspondiente del valor dado por f .

Si bien SVM es un modelo de clasificación binario, este se puede extender a un modelo multi-clase. LIBSVM implementa el enfoque uno contra uno. Si k es el número de clases, entonces se construyen $k(k-1)/2$ clasificadores y cada uno entrena todos los datos para 2 clases. En la clasificación se usa una estrategia de votación: cada clasificación binaria es una votación donde los votos pueden ser emitidos para todos los ejemplos x_i . Al final cada ejemplo es asignado a la clase con el máximo número de votos.

Los mismos creadores de LIBSVM exponen en un tutorial (Hsu y col., 2003) una serie de pasos para llevar a cabo un mejor proceso de *Machine Learning* utilizando SVM, destacando entre ellos la normalización de los datos numéricos y búsqueda de los parámetros óptimos para el modelo.

Para comenzar con la etapa de entrenamiento y validación, los datos fueron exportados desde la base de datos en formato CSV, y luego importados a Matlab. Una vez los datos estuvieron cargados en Matlab estos fueron normalizados entre valores -1 y 1, mediante un script (anexo 3.1). Inmediatamente, con otro script (anexo 3.2), estos datos normalizados fueron divididos por tipo de información para realizar las distintas pruebas y evaluar el desempeño al integrar diversos tipos de información.

Posteriormente, uno de los puntos más importantes fue la búsqueda de los parámetros óptimos:

- 1) C : es cual corresponde al costo de incluir más vectores de soporte al modelo.
- 2) γ : el cual es un parámetro de regularización de la función *kernel* de tipo gaussiana.

Esta búsqueda fue realizada para cada tipo de información y para las diferentes combinaciones de ellas:

- 1) Evolutiva
- 2) Geométrica
- 3) Fisicoquímica
- 4) Evolutiva + Geométrica (Evo+Geo)
- 5) Evolutiva + Fisicoquímica (Evo+Fis)
- 6) Geométrica + Fisicoquímica (Geo+ Fis)
- 7) Evolutiva + Geométrica + Fisicoquímica (Evo+Geo+Fis)

Fue construido un script (anexo 3.3), para probar diferentes combinaciones de los parámetros (C , γ), y entregar como salida la combinación de estos 2 parámetros con mayor exactitud. Cada exactitud era obtenida mediante validación cruzada de 10 iteraciones. En la validación cruzada de 10 iteraciones los datos se dividen en 10 subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. El proceso de validación cruzada es repetido durante 10 iteraciones, con cada uno de los subconjuntos de datos de prueba. Finalmente se calcula la media aritmética de los resultados de cada iteración para obtener un único desempeño.

Una vez encontrados estos parámetros óptimos, se realizó la etapa de entrenamiento y testeo de los modelos para cada tipo de información, mediante validación cruzada de 10 iteraciones, la cual cada fue implementada con un script (anexo 3.4), ya que la validación cruzada implementada en las librerías de LIBSVM sólo entrega la exactitud global y no las tasas por clase. De la validación cruzada fueron obtenidas distintas medidas de desempeño:

$$✓ \text{ Sensibilidad} = \text{TPR} = \text{TP}/(\text{TP}+\text{FN}) \quad (6)$$

$$✓ \text{ Selectividad} = \text{FPR} = \text{FP}/(\text{FP}+\text{TN}) \quad (7)$$

$$✓ \text{ Exactitud} = (\text{TP}+\text{TN})/(\text{TP}+\text{FP}+\text{FN}+\text{TN}) \quad (8)$$

Donde:

✓ TPR: *True Positives Rate*

✓ FPR: *False Positives Rate*

✓ TP: *True Positives*

✓ TN: *True Negatives*

✓ FP: *False Positives*

✓ FN: *False Negatives*

El aumento de la cantidad de atributos utilizados en un modelo predicción disminuye el desempeño del mismo (Bishop, 2007). Se consideró necesario reducir la dimensionalidad del vector de atributos, para esto se utilizó una estrategia de 2 etapas:

- 1) Eliminación de atributos correlacionados: esta metodología consistió en calcular una matriz de coeficientes de correlación de Pearson para los atributos, todos contra todos, y eliminar aquellos con alta correlación.
- 2) Eliminación de atributos que generan ruido al modelo: se midió la exactitud del modelo al remover cada uno de los atributos, uno a la vez. Se descartaron aquellos que producían un aumento de exactitud cuando no eran considerados en el modelo.

4.- Comparación de SVM vs otros modelos supervisados

Finalmente se llevó a cabo una comparación del modelo basado en SVM contra modelos de aprendizaje supervisados implementados en WEKA (Hall y col., 2009) dejando todos los parámetros por defecto. Los modelos a comparar fueron:

- 1) Árboles de Decisión (DT): `weka.classifiers.trees.J48`
- 2) Clasificador Bayesiano Ingenuo (NBC): `weka.classifiers.bayes.NaiveBayes`
- 3) Regresión logística (LR): `weka.classifiers.functions.SimpleLogistic`

RESULTADOS

1.- Set de datos no redundante

La tabla 2 muestra la cantidad cadenas de proteínas y sitios de unión utilizados para el estudio. La búsqueda en PDB de estructuras que contuvieran Ca, Mg o Zn, usando el parámetro de búsqueda avanzada *Chemical ID* encontró un total de 17.095 estructuras. Del conteo realizado para identificar sólo las cadenas que unieran algunos de estos tres metales resultó un total de 10.367 Cadenas-M para Ca, 13.334 Cadenas-M para Mg y 12.659 Cadenas-M para Zn. Posteriormente, la eliminación de cadenas redundantes y de baja calidad utilizando PISCES entregó un total de 424 Cadenas-M no redundante para Ca, 612 Cadenas-M no redundante para Mg y 500 Cadenas-M no redundante para Zn.

La extracción de los sitios de unión usando VMD arrojó un total de 452 sitios para Ca, 421 para Mg y 531 para Zn. Sin embargo, en este proceso se detectaron estructuras de sitios de unión que contenían residuos con más de una conformación en su cadena lateral. Luego de la eliminación de estos sitios de unión que representaban ruido para el modelo, se obtuvo un total de 1148 sitios de unión, en donde 383 eran de Ca, 331 eran de Mg y 434 eran de Zn.

Tabla 2. Cantidad de Cadenas-M, Cadenas-M-NR (no redundantes) y sitios de unión a metales obtenidos para el estudio.

Metal	Cadenas-M	Cadenas-M-NR	Sitios de unión a metal
Ca	10.367	424	383 (452)
Mg	13.334	612	331 (421)
Zn	12.659	500	434 (531)

Los valores entre paréntesis representan la cantidad de sitios de unión antes de filtrar los sitios con residuos con más de una conformación.

2.- Atributos

Los atributos calculados por distintas metodologías, fueron agrupados en 9 tablas según su tipo de información y metodología de estimación.

2.1.- Atributos Evolutivos

En primer lugar, para poder estimar los puntajes evolutivos se tuvo que construir Pseudo-PSSMs para cada tipo de sitio de unión a metal. Estas permiten conocer la preferencia de estos 3 metales para con los aminoácidos. En las figuras 7, 8 y 9 se muestra la distribución de aminoácidos en los sitios de unión a Ca, Mg y Zn, respectivamente. Se puede observar que para el Ca existe una clara preferencia por ácido aspártico (D) y ácido glutámico (E) en los 4 residuos más cercanos al metal. Para el Mg existe una tendencia similar a la del Ca, pero con una diferencia en el cuarto residuo, donde el más frecuente es la lisina (K). En los sitio de unión a Zn la tendencia diverge con respecto al Ca y Mg, aquí los aminoácidos más frecuentes son la histidina (H), en la primera y segunda posición, y la cisteína (C), en la tercera y cuarta posición.

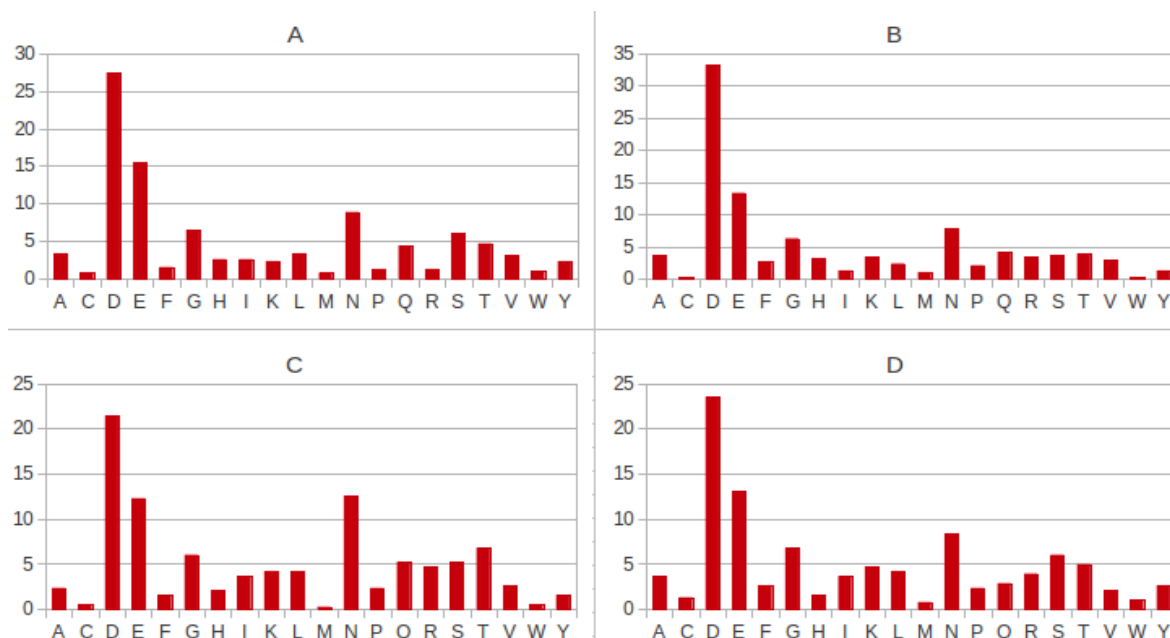


Figura 7. Distribución de aminoácidos para los sitios de unión a Calcio. (A) Frecuencias del aminoácido más cercano. (B) Frecuencias del segundo aminoácido más cercano. (C) Frecuencias del tercer aminoácido más cercano. (D) Frecuencias del cuarto aminoácido más cercano.

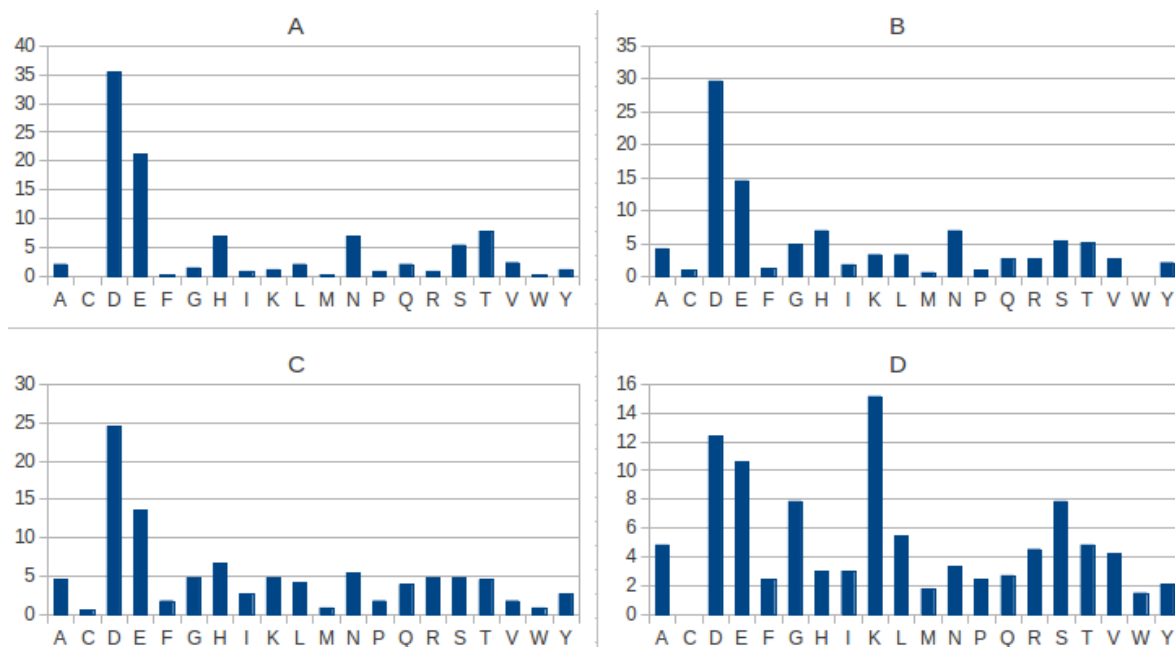


Figura 8. Distribución de aminoácidos para los sitios de unión a Magnesio. (A) Frecuencias del aminoácido más cercano. (B) Frecuencias del segundo aminoácido más cercano. (C) Frecuencias del tercer aminoácido más cercano. (D) Frecuencias del cuarto aminoácido más cercano.

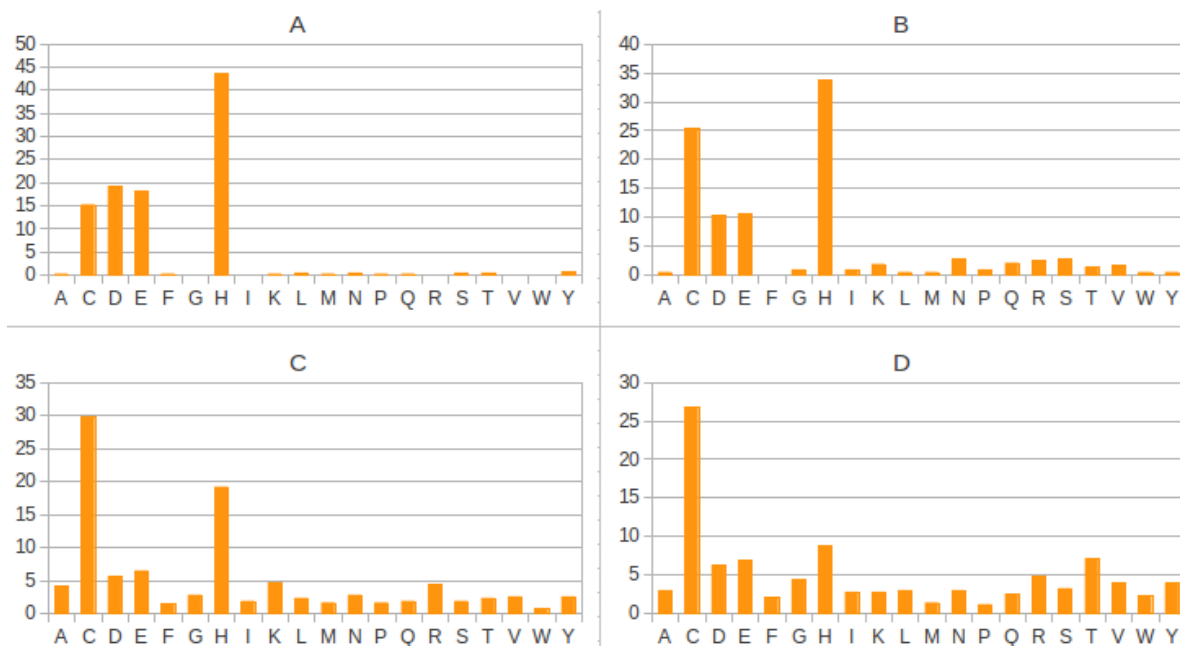


Figura 9. Distribución de aminoácidos para los sitios de unión a Zinc. (A) Frecuencias del aminoácido más cercano. (B) Frecuencias del segundo aminoácido más cercano. (C) Frecuencias del tercer aminoácido más cercano. (D) Frecuencias del cuarto aminoácido más cercano.

En la tabla 3 se pueden observar los valores promedio y desviaciones estándar obtenidos para los 3 atributos evolutivos. Para cada tipo de metal se obtuvo valores distintos, pero el Score Ca-PSSM y Score Mg-PSSM, parecen entregar valores muy similares para los tres tipos de sitios de unión

Tabla 3. Valores promedio y desviaciones estándar de atributos evolutivos para cada metal.

	Ca		Mg		Zn	
Score	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
Zn-PSSM	18,54	10,46	22,85	13,25	54,63	23,88
Mg-PSSM	43,81	20,05	48,21	21,84	29,34	19,87
Ca-PSSM	44,75	18,99	43,41	20,50	23,57	18,14

2.2.- Atributos Geométricos

La tabla 4 muestra los valores promedio y desviaciones estándar para los atributos geométricos basados en distancias y la tabla 5 para los atributos geométricos basados en ángulos. Visualmente, es posible identificar que el Zn, en comparación al Ca y al Mg, forma interacciones más cortas con las cadenas laterales de los cuatro aminoácidos más cercanos (Ver tabla 4, distancias M-R1, M-R2, M-R3 y M-R4). Además, en el atributo Distancia M-R1, definido como la primera interacción del metal con el sitio de unión, se puede observar una clara diferencia para estos tres metales, 2,36 Å para Ca, 2,18 Å para Mg y 2,09 Å para Zn.

Tabla 4. Valores promedio y desviaciones estándar de atributos geométricos basados en distancias para cada metal. Valores en Å

	Ca		Mg		Zn	
Distancia	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
M-R1	2,36	0,18	2,18	0,23	2,09	0,17
M-R2	2,84	0,84	2,96	0,96	2,63	0,86
M-R3	3,45	1,20	3,70	1,13	3,29	1,28
M-R4	4,03	1,39	4,46	0,97	4,02	1,45
M-Ca1	4,75	0,74	5,08	0,98	5,37	0,98
M-Ca2	5,15	1,15	5,52	1,35	5,51	1,12
M-Ca3	5,50	1,36	5,93	1,50	5,77	1,47
M-Ca4	5,95	1,58	6,57	1,74	6,11	1,60
R1-R2	3,76	0,89	3,64	0,88	3,62	0,60
R1-R3	3,95	1,05	4,11	1,11	3,98	0,88
R1-R4	4,33	1,22	4,65	1,22	4,47	1,09
R2-R3	4,22	1,31	4,35	1,37	4,15	1,15
R2-R4	4,49	1,48	4,86	1,60	4,68	1,41
R3-R4	4,72	1,60	5,37	1,75	4,84	1,59
Ca1-Ca2	6,73	1,77	6,92	2,07	7,34	1,97
Ca1-Ca3	6,52	2,20	6,83	2,35	7,35	2,17
Ca1-Ca4	6,63	2,35	7,36	2,92	7,41	2,52
Ca2-Ca3	6,83	2,19	6,95	2,57	7,37	2,27
Ca2-Ca4	7,09	2,50	7,61	2,87	7,50	2,47
Ca3-Ca4	7,22	2,36	7,95	2,95	7,53	2,59

Tabla 5. Valores promedio y desviaciones estándar de atributos geométricos basados en ángulos para cada metal. Valores en grados.

	Ca		Mg		Zn	
Ángulo	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
R1-M-R2	96,88	33,69	91,89	30,35	101,96	21,58
R1-M-R3	87,58	37,41	86,51	32,37	94,53	26,48
R1-M-R4	82,76	34,16	83,84	37,25	90,17	32,36
R2-M-R3	89,67	31,46	87,15	34,46	95,09	27,25
R2-M-R4	85,57	35,53	83,16	35,28	92,12	31,27
R3-M-R4	86,46	34,41	88,11	35,00	91,53	32,21
Ca1-M-Ca2	89,93	31,40	83,58	28,34	88,75	30,36
Ca1-M-Ca3	82,78	35,07	78,44	31,77	87,75	35,28
Ca1-M-Ca4	78,91	34,44	79,88	35,78	85,38	38,06
Ca2-M-Ca3	84,63	33,47	77,25	31,73	87,39	34,74
Ca2-M-Ca4	83,76	34,98	81,00	33,96	86,54	37,04
Ca3-M-Ca4	82,82	33,76	83,13	35,85	85,00	36,33
M-R1-Ca1	129,15	17,08	137,29	20,83	132,35	22,61
M-R2-Ca2	126,79	19,37	127,43	21,03	128,33	21,06
M-R3-Ca3	121,68	23,19	121,17	24,59	123,65	21,94
M-R4-Ca4	121,60	24,14	119,32	26,88	119,88	21,63

2.3.- Atributos Fisicoquímicos

Por último, los valores de los atributos fisicoquímicos fueron resumidos estadísticamente en las tablas 6, 7, 8, 9, 10 y 11. En la tabla 6 se muestran las hidrofobicidades para los 4 residuos más cercanos al metal. En ella se puede observar que el cuarto residuo más cercano tiende a ser el más hidrofóbico para los tres metales. En los atributos basados en composición atómica, expuestos en la tabla 7, se puede apreciar que el porcentaje de átomos de oxígeno en la capa 3 es mucho mayor para el Ca y el Mg, 88% y 80% respectivamente, en comparación al Zn que sólo promedia un 21%. En el caso del conteo de átomos por capas resumidos en la tabla 8, se evidencia un aumento de átomos a medida que se cambia a una capa más lejana del metal. En los atributos basados en energía presentados en la tabla 9, la energía total de sitio muestra una clara diferencia entre los metales, donde para el Ca y el Mg son valores positivos de 10,18 Kcal/mol y 11,73 Kcal/mol, respectivamente, y para el Zn es -0,57 Kcal/mol. El área accesible al solvente, mostrada en la tabla 10 parece ser uniforme para los tres tipos de metales, los valores son 1356,68, 1427,96 y 1415,62, para Ca, Mg y Zn, respectivamente. Por último, en la tabla 11 se muestra la composición aminoacídica del sitio de unión. En ella se puede observar preferencia por los aminoácidos polares neutros para los tres tipos de metales, pero analizando sólo los aminoácidos polares con cargas, existe una clara diferencia de Ca y Mg vs el Zn, ya que el Ca y el Mg prefieren los polares negativos, mientras que el Zn prefiere los polares positivos.

Tabla 6. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en hidrofobicidades para cada metal.

Hidrofobicidad	Ca		Mg		Zn	
	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
1er aa escala 1	-1,85	2,53	-2,47	2,04	-2,34	2,18
1er aa escala 2	-0,71	0,85	-0,98	0,76	-0,98	0,71
1er aa escala 3	1,93	1,38	2,30	1,24	2,07	1,14
2do aa escala 1	-2,13	2,40	-2,03	2,45	-1,45	2,74
2do aa escala 2	-0,79	0,78	-0,79	0,78	-0,66	0,74
2do aa escala 3	2,14	1,34	2,04	1,35	1,58	1,26
3er aa escala 1	-1,88	2,61	-1,92	2,56	-0,51	2,96
3er aa escala 2	-0,65	0,78	-0,70	0,83	-0,36	0,76
3er aa escala 3	1,87	1,33	1,93	1,36	1,13	1,29
4to aa escala 1	-1,68	2,66	-1,40	2,81	-0,23	2,90
4to aa escala 2	-0,62	0,86	-0,53	0,83	-0,23	0,81
4to aa escala 3	1,81	1,40	1,59	1,36	0,97	1,28

Tabla 7. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en composición atómica para cada metal.

Porcentaje	Ca		Mg		Zn	
	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
C capa 2	0,00	0,00	0,30	5,50	0,12	2,40
O capa 2	0,52	7,22	14,85	35,41	18,58	37,55
N capa 2	0,26	5,11	1,16	10,02	7,79	24,65
S capa 2	0,26	5,11	0,00	0,00	0,46	6,78
C capa 3	8,34	13,76	9,01	20,60	27,91	26,46
O capa 3	88,41	20,20	80,02	33,93	20,50	26,50
N capa 3	2,61	14,41	7,12	21,02	26,71	28,89
S capa 3	0,13	2,55	0,15	2,75	24,39	37,58
C capa 4	75,95	24,11	61,56	23,78	70,06	30,88
O capa 4	11,49	15,02	24,41	23,52	13,51	24,01
N capa 4	7,99	12,18	13,02	16,41	9,35	14,93
S capa 4	0,04	0,72	0,04	0,77	0,06	0,95
C capa 5	56,80	13,25	57,34	14,91	63,24	14,33
O capa 5	19,10	14,78	22,35	14,72	11,49	11,62
N capa 5	23,18	11,74	19,44	12,82	24,16	11,77
S capa 5	0,01	0,26	0,08	0,76	0,28	1,86
C capa 6	66,39	13,32	63,60	14,36	61,62	13,84
O capa 6	16,57	10,89	16,63	10,37	17,88	10,38
N capa 6	15,78	9,89	18,54	11,99	19,02	10,93
S capa 6	0,25	1,40	0,25	1,48	0,26	1,32
C capa 7	55,74	12,99	57,81	12,13	60,51	11,42
O capa 7	24,99	11,26	22,36	10,81	20,83	9,54
N capa 7	17,66	10,96	18,46	9,44	17,13	9,50
S capa 7	0,54	1,99	0,29	1,55	0,40	1,64

Tabla 8. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en conteo de átomos para cada metal.

	Ca		Mg		Zn	
Total átomos	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
Capa 2	0,01	0,10	0,19	0,47	0,36	0,68
Capa 3	3,81	2,18	2,23	1,31	3,74	1,57
Capa 4	4,33	2,85	4,46	2,51	4,37	2,52
Capa 5	13,18	6,66	10,27	5,13	9,65	4,27
Capa 6	14,14	6,59	13,48	5,74	15,41	8,20
Capa 7	16,53	7,39	17,79	7,14	19,25	7,88

Tabla 9. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en energías para cada metal. Valores en Kcal/mol.

	Ca		Mg		Zn	
Energía	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
Energía total	10,18	6,46	11,73	6,03	-0,57	7,63
<i>Backbone Hbond</i>	-2,39	1,79	-2,19	1,75	-2,55	1,96
<i>Sidechain Hbond</i>	-2,00	1,84	-1,99	1,88	-1,61	1,56
Van der Waals	-5,45	3,73	-4,72	3,04	-6,69	4,57
Electrostática	-1,38	1,61	-0,91	1,60	-0,87	1,29
Solvatación polar	18,56	10,19	15,14	8,05	18,01	9,25
Solvatación hidrofóbica	-6,08	4,26	-5,24	3,53	-7,93	5,65
Van der Waals <i>clashes</i>	1,51	1,76	1,24	1,53	1,50	1,57
Entropía de <i>sidechain</i>	4,24	2,43	4,52	3,02	5,53	2,97
Entropía de <i>mainchain</i>	9,91	5,51	7,96	4,27	9,52	5,85
<i>Torsional clash</i>	1,05	1,07	1,06	0,99	1,20	1,08
<i>Backbone clash</i>	2,93	2,14	2,49	1,86	2,96	2,58
Enlaces parciales	-7,76	5,17	-3,24	2,34	-16,98	10,21
Energía de ionización	0,03	0,10	0,08	0,20	0,29	0,26
Número de residuos	99,54	93,77	126,19	105,00	99,46	123,39

Tabla 10. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en área accesible al solvente para cada metal. Valores en Å²

	Ca		Mg		Zn	
Atributo	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
SASA	1356,68	368,82	1427,96	399,90	1415,62	418,76

Tabla 11. Valores promedio y desviaciones estándar de atributos fisicoquímicos basados en composición aminoacídica para cada metal.

	Ca		Mg		Zn	
Porcentaje de residuos	Promedio	Desv. Estándar	Promedio	Desv. Estándar	Promedio	Desv. Estándar
No polares	26,00	13,97	24,06	14,30	23,40	13,26
Polares neutros	38,37	16,66	37,63	17,00	38,03	17,11
Polares negativos	23,58	12,57	21,99	13,19	16,33	12,67
Polares positivos	10,84	10,24	15,10	12,39	20,97	11,51

3.- Desempeño de los modelos de predicción basados en SVM

El proceso de predicción y discriminación de sitios de unión a Zn, Mg y Ca mediante SVM fue testeado en siete distintos escenarios, los cuales usaron distintos tipos de información. Los parámetros óptimos C y γ fueron obtenidos en cada caso utilizando validación cruzada de 10 iteraciones (Valores óptimos pueden ser revisado en el anexo 5).

En la figura 10 se presentan los resultados de exactitud de los distintos modelos generados. Para el basado en información evolutiva fue un 66,08%, para el basado en información geométrica fue 69,69%, para el basado en información fisicoquímica fue 83,84%, para el basado en la combinación de información evolutiva y geométrica fue de 74,56%, para el basado en la combinación de evolutiva y fisicoquímica fue 84,57%, para el basado en la combinación de información geométrica y fisicoquímica fue de 88,66% y finalmente para el que integraba los tres tipos de información la exactitud fue de 88,23%. Cabe destacar de estos resultados que, con la combinación de distintos tipos de información las exactitudes en general tienden a aumentar.

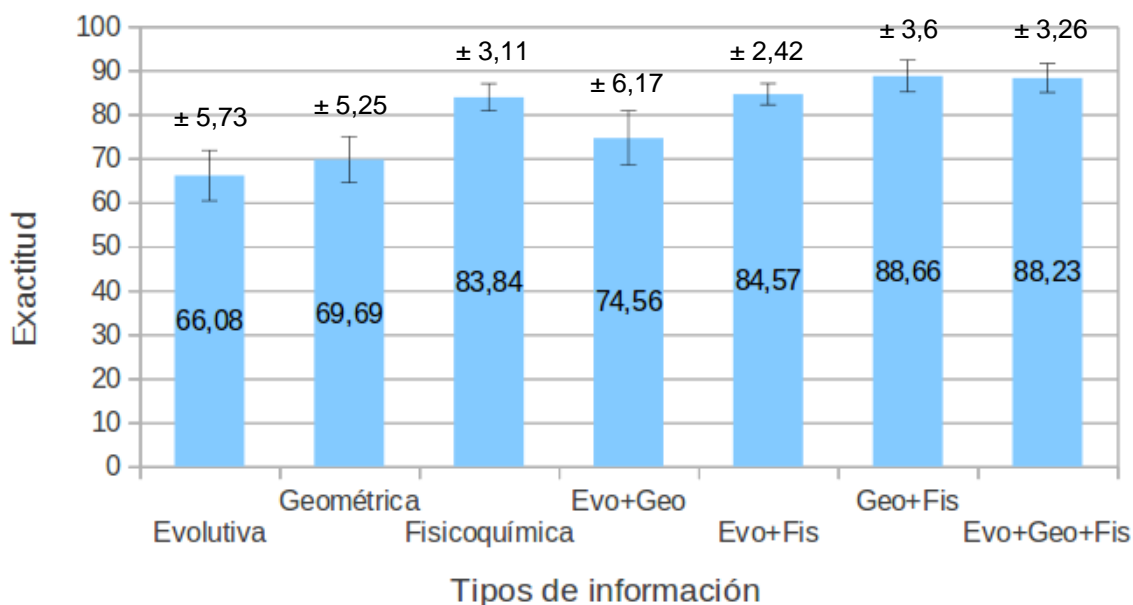


Figura 10. Exactitudes obtenidas usando SVM y diversos tipos de información.

En la tabla 12 se muestra las tasas de verdaderos positivos y falsos positivos para cada clase. Los modelos con mayor capacidad de discriminación entre sitios de unión fueron aquellos basados en información Geo+Fis e información Evo+Geo+Fis, logrando tasas de TP sobre el 0,85 y FP positivos bajo 0,07. En cambio la información evolutiva y geométrica, ambas utilizadas individualmente, tienen las tasas de TP y FP más bajas, especialmente para discriminar los sitios de unión a Mg.

Tabla 12. Medidas de desempeño para la discriminación entre sitios de unión a metales usando SVM.

SVM						
Información	TPR Ca	TPR Mg	TPR Zn	FPR Ca	FPR Mg	FPR Zn
Evolutiva	0,67 ± 0,07	0,48 ± 0,09	0,79 ± 0,08	0,21 ± 0,07	0,19 ± 0,05	0,11 ± 0,03
Geométrica	0,76 ± 0,06	0,51 ± 0,11	0,78 ± 0,06	0,16 ± 0,05	0,15 ± 0,04	0,14 ± 0,04
Fisicoquímica	0,80 ± 0,07	0,80 ± 0,07	0,90 ± 0,05	0,09 ± 0,03	0,10 ± 0,04	0,04 ± 0,02
Evo+Geo	0,80 ± 0,08	0,57 ± 0,08	0,84 ± 0,09	0,15 ± 0,05	0,14 ± 0,06	0,09 ± 0,05
Evo+Fis	0,80 ± 0,06	0,81 ± 0,06	0,91 ± 0,04	0,09 ± 0,02	0,10 ± 0,02	0,04 ± 0,02
Geo+Fis	0,89 ± 0,05	0,86 ± 0,06	0,90 ± 0,06	0,07 ± 0,03	0,06 ± 0,03	0,04 ± 0,02
Evo+Geo+Fis	0,88 ± 0,06	0,85 ± 0,06	0,91 ± 0,05	0,07 ± 0,03	0,07 ± 0,03	0,03 ± 0,02

Tasas de Verdaderos Positivos (TPR) y Falsos Positivos (FPR) para cada clase obtenidas con SVM, utilizando diversos tipos de información y sus distintas combinaciones.

Un proceso de reducción de dimensionalidad del vector de atributos, fue realizado con el fin de optimizar el modelo basado en SVM que integraba los 3 tipos de información. La figura 11 muestra la distribución de los coeficientes de correlación de Pearson calculados para cada par de atributos. En ella se aprecia que, en la mayoría de los casos, los coeficientes de correlación son cercanos a 0. Sin embargo, se pudo identificar 3 patrones de tipos de atributos correlacionados. El primer patrón fue identificado en los atributos evolutivos, donde se obtuvo que el Score Mg-PSSM y el Score Ca-PSSM estaban altamente correlacionado con un coeficiente de Pearson de 0,94 (Ver tabla 13). Otro patrón de atributos correlacionados fue encontrado en los atributos basados en hidrofobicidades. En la tabla 14 se puede observar que existe una alta correlación entre las tres escalas de hidrofobicidad, pero es más alta entre las escalas 1 y 3. El tercer patrón de atributos correlacionados fue encontrado en los de tipo energético. En la tabla 16 se puede observar que existe una alta correlación entre las energías de Van der Waals, solvatación polar, solvatación hidrofóbica y la entropía *mainchain*. Cabe destacar además que de entre estos cuatro atributos correlacionados, la entropía de la cadena principal es el que tiene un menor promedio de correlación con todos los atributos del modelo. Con esta metodología 12 atributos fueron eliminados.

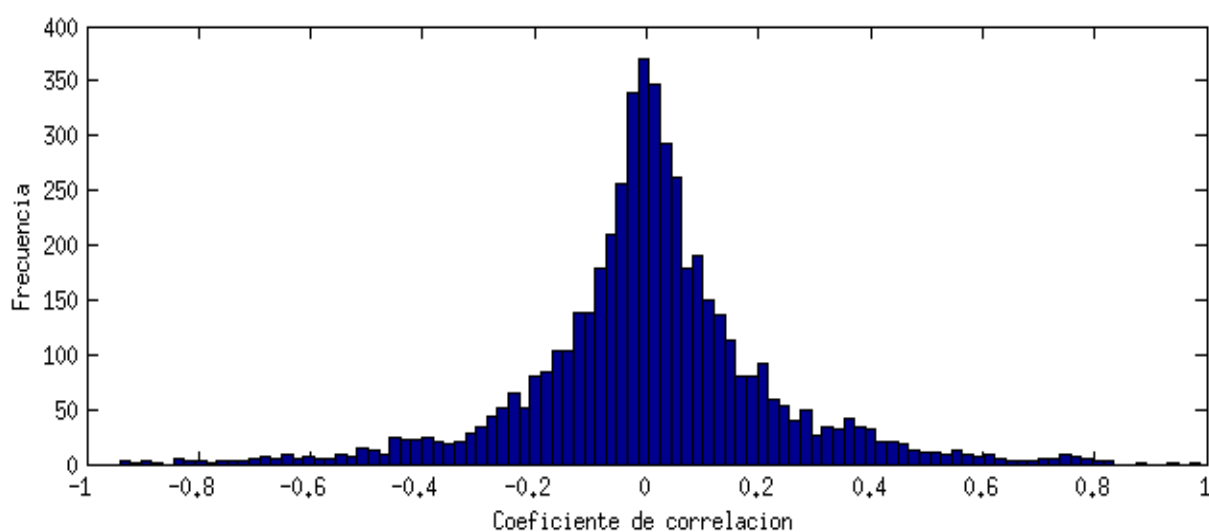


Figura 11. Distribución de los coeficientes de correlación entre atributos del modelo.

Tabla 13. Correlaciones entre atributos Evolutivos basados en Pseudo-PSSM.

	Score Zn-PSSM	Score Mg-PSSM	Score Ca-PSSM	Todos los att.
Score Zn-PSSM	1,00	-0,31	-0,43	0,04
Score Mg-PSSM	-0,31	1,00	0,94	0,01
Score Ca-PSSM	-0,43	0,94	1,00	0,00

Tabla 14. Correlaciones entre atributos Fisicoquímicos basados en Hidrofobicidades.

	Hidro del 1er aa escala 1	Hidro del 1er aa escala 2	Hidro del 1er aa escala 3	Todos los att.
Hidro del 1er aa escala 1	1,00	0,75	-0,89	-0,01
Hidro del 1er aa escala 2	0,75	1,00	-0,80	-0,02
Hidro del 1er aa escala 3	-0,89	-0,80	1,00	0,01
	Hidro del 2do aa escala 1	Hidro del 2do aa escala 2	Hidro del 2do aa escala 3	Todos los att.
Hidro del 2do aa escala 1	1,00	0,77	-0,91	-0,01
Hidro del 2do aa escala 2	0,77	1,00	-0,83	-0,01
Hidro del 2do aa escala 3	-0,91	-0,83	1,00	0,01
	Hidro del 3er aa escala 1	Hidro del 3er aa escala 2	Hidro del 3er aa escala 3	Todos los att.
Hidro del 3er aa escala 1	1,00	0,75	-0,92	-0,01
Hidro del 3er aa escala 2	0,75	1,00	-0,84	-0,01
Hidro del 3er aa escala 3	-0,92	-0,84	1,00	0,01

Tabla 15. Correlaciones entre atributos Fisicoquímicos basados en Energías.

	Van der Waals	Solvatación Polar	Solvatación hidrofóbica	Entropía <i>mainchain</i>	Todos los att.
Van der Waals	1,00	-0,94	0,99	-0,89	-0,06
Solvatación Polar	-0,94	1,00	-0,90	0,87	0,06
Solvatación hidrofóbica	0,99	-0,90	1,00	-0,86	-0,06
Entropía <i>mainchain</i>	-0,89	0,87	-0,86	1,00	0,04

La segunda etapa de reducción de dimensionalidad consistió en eliminar atributos que generaban ruido al el modelo, para esto se validó el modelo sacando todos los atributos, uno a la vez, y observar cuales atributos al ser eliminados del modelo le producían un aumento de exactitud a la tarea de clasificación. En la tabla 16 se muestran la lista de atributos que producían ruido para el modelo con el aumento de exactitud en la predicción, cuando cada atributo era removido del modelo. En etapa 13 atributos fueron eliminados.

Tabla 16. Atributos que representan ruido para SVM con el aumento de exactitud producido al ser removidos del modelo.

Atributo	Aumento del % Exactitud
Porcentaje N capa 4	0,43
Score Zn-PSSM	0,18
Ángulo R1-M-R4	0,18
Ángulo R3-M-R4	0,09
Distancia R2-R4	0,09
Porcentaje O capa 4	0,09
Entropía <i>sidechain</i>	0,09
Distancia M-C α 4	0,08
Distancia R1-R4	0,08
Distancia R3-R4	0,08
Porcentaje N capa 2	0,08
Hidro del 3er aa escala 2	0,08
Total átomos capa 2	0,01

Posterior a la reducción de dimensionalidad del vector de atributos se midió la nuevamente el desempeño del modelo de predicción. En la tabla 17 se muestran las medidas de desempeño de SVM. Cuando se eliminaron los atributos correlacionados (SC) la exactitud global del modelo aumentó de 88,23% a 89,04%. Luego al eliminar los atributos que representaban ruido para el modelo (SCSR), se alcanzó un desempeño de 90,41% de exactitud. Además, las tasas de TP también se vieron beneficiadas, logrando 0,89 para el Ca y el Mg y 0,93 para el Zn.

Tabla 17. Medidas de desempeño para SVM con reducción de dimensionalidad.

SVM con reducción de dimensionalidad							
Información	Exactitud	TPR Ca	TPR Mg	TPR Zn	FPR Ca	FPR Mg	FPR Zn
Evo+Geo+Fis	88,23 \pm 3,26	0,88 \pm 0,06	0,85 \pm 0,06	0,91 \pm 0,05	0,07 \pm 0,03	0,07 \pm 0,03	0,03 \pm 0,02
Evo+Geo+Fis SC	89,04 \pm 3,31	0,87 \pm 0,05	0,86 \pm 0,06	0,92 \pm 0,05	0,06 \pm 0,03	0,07 \pm 0,03	0,03 \pm 0,02
Evo+Geo+Fis SCSR	90,41 \pm 2,27	0,89 \pm 0,05	0,89 \pm 0,06	0,93 \pm 0,05	0,05 \pm 0,03	0,05 \pm 0,03	0,03 \pm 0,02

Tasa de Verdaderos Positivos (TPR) y Falsos Positivos (FPR) para cada clase. Evo+Geo+Fis: modelo que integra los tres tipos de información. Evo+Geo+Fis SC: modelo que integra los tres tipos de información sin atributos correlacionados. Evo+Geo+Fis SCSR: modelo que integra los tres tipos de información sin atributos correlacionados y sin atributos que generan ruido al modelo.

4.- SVM contra otros modelos de aprendizaje supervisado

Finalmente, se realizó una comparación del modelo basado en SVM, versus modelos de aprendizaje supervisado basados en otros paradigmas de *Machine Learning*. En la figura 12 se muestran los valores de exactitud alcanzada por DT, NBC, LR y SVM. De esta comparación se puede inferir que, el modelo con menor desempeño es NBC logrando sólo un 72,39% de exactitud. DT alcanzó un desempeño de 83,88%. Por otro lado LR obtuvo un desempeño de 91.03% de exactitud, levemente superior al 90,41% logrado por SVM.

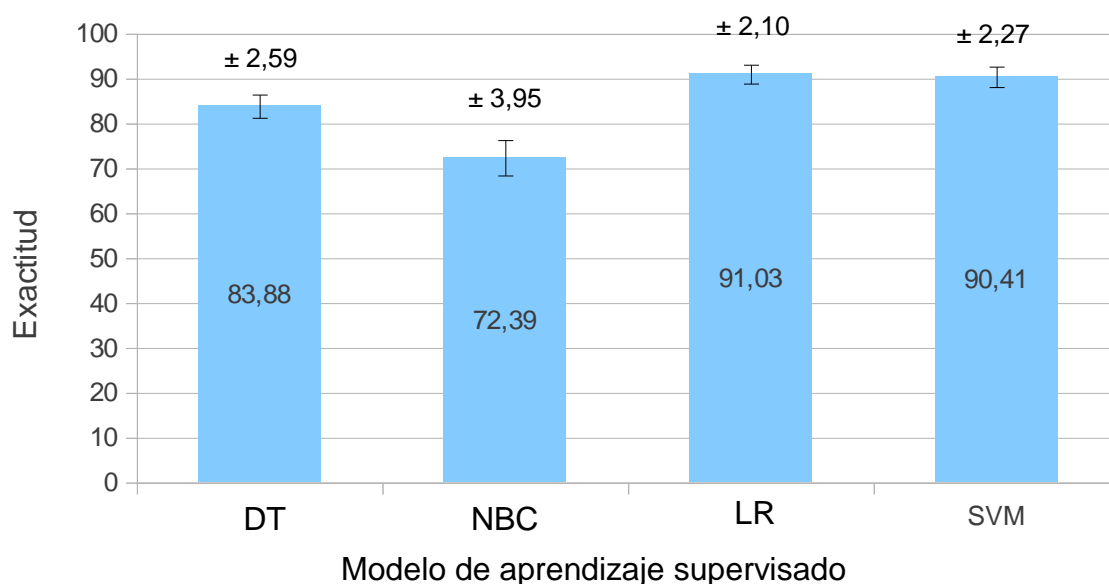


Figura 12. Exactitudes logradas por Árboles de Decisión (DT), Clasificador Bayesiano Ingenuo (NBC), Regresión Logística (LR) y Máquinas de Vectores de Soporte (SVM).

Finalmente, la tabla 18 muestra la tasas de TP y FP logrados por los cuatro modelos de aprendizaje supervisado. En general, las tasas de verdaderos positivos de Zn son las más altas de las tres clases, a excepción de NBC en el cual la tasa de verdaderos positivos de Ca es la mayor. Por otro lado, se puede observar que DT y NBC, son los dos modelos con menor desempeño, y además tienen bajas tasas de TP para el Mg.

Tabla 18. Medidas de desempeño por clase para distintos moldes de aprendizaje supervisado.

Modelo supervisado	TPR Ca	TPR Mg	TPR Zn	FPR Ca	FPR Mg	FPR Zn
DT	0,83 ± 0,05	0,76 ± 0,08	0,91 ± 0,04	0,09 ± 0,03	0,10 ± 0,03	0,04 ± 0,03
NBC	0,78 ± 0,07	0,60 ± 0,09	0,77 ± 0,07	0,17 ± 0,03	0,20 ± 0,05	0,04 ± 0,03
LR	0,90 ± 0,06	0,90 ± 0,05	0,93 ± 0,02	0,05 ± 0,02	0,06 ± 0,03	0,02 ± 0,01
SVM	0.89 ± 0.05	0.89 ± 0.06	0.93 ± 0.05	0.05 ± 0.03	0.05 ± 0.03	0.03 ± 0.02

Las tasas de Verdaderos Positivos (TPR) y Falsos Positivos (FPR) para cada clase logradas por Árboles de Decisión (DT), Clasificador Bayesiano Ingenuo (NBC), Regresión Logística (LR) y Máquinas de Vectores de Soporte (SVM).

DISCUSIÓN

Identificar los sitios de unión a metales es de relevancia para poder comprender el funcionamiento de una proteína o para ayudar en etapas del diseño de nuevos fármacos. En este estudio se ha desarrollado un modelo de predicción y discriminación de sitios de unión a metales utilizando SVM, en el cual además se integraron diferentes tipos de información. Con este modelo se logró sobre un 90% de exactitud en la predicción mejorando la capacidad discriminativa de estudios anteriores (Bordner, 2008) (Ebert y Altman, 2008).

La creación del set de datos no redundante de estructuras de proteínas, con alta calidad y que cumplieran con los criterios mencionados en las secciones anteriores, permitió reducir considerablemente el tamaño inicial del set de datos disponible, aproximadamente 20 veces (Ver tabla 2). Estos números dejan en evidencia la alta redundancia en la base de datos PDB, además de establecer la importancia de esta etapa para todo tipo de análisis basados en *Machine Learning*. La disminución de la cantidad de cadenas proteicas puede ser atribuida a los dos principales filtros utilizados en PISCES, el porcentaje de identidad de secuencia para el agrupamiento y el rango de resolución para las estructuras, en donde el primero tiene su efecto en la eliminación de cadenas redundantes y el segundo en la eliminación de cadenas de baja calidad. Se utilizaron estructuras entre 0 y 2 Å de resolución. Este rango puede ser considerado estricto y ocasionar la exclusión de estructuras importantes para el modelo. Por lo tanto, es preciso mencionar que para futuras investigaciones el margen de resolución puede ser ampliado y determinar si provoca algún cambio en la exactitud del modelo de predicción.

En relación a la extracción de sitios de unión a metales, esta permitió obtener 383 sitios para Ca, 331 sitios para Mg y 434 sitios para Zn. Las cantidades por clases son similares a los utilizados en estudios anteriores (Brylinski y Skolnick, 2011) (Bordner, 2008). Se ha descrito que modelos de predicción basados en SVM se ven afectados negativamente por el desbalance de clases (Bishop, 2007). En este

estudio la cantidad de sitios de unión utilizados no mostró un problema real de clases desbalanceadas, ya que el modelo predice de manera eficiente y equilibrada para todas las clases. Además, no se observan diferencias significativas en las tasas de falsos positivos. Un problema detectado, no reportado por trabajos anteriores, es la presencia de residuos con más de una conformación en los sitios de unión, es decir sus cadenas laterales tienen más de una posición. Este problema puede afectar a los atributos de tipo geométricos o fisicoquímicos tales como, distancias, ángulos, conteo de átomos y energías. Para evitar este problema estos sitios fueron descartados.

Con respecto a la estimación de atributos evolutivos, geométricos y fisicoquímicos, esta mostró diferencias entre los diferentes tipos de sitios de unión a metales. De las frecuencias obtenidas en la construcción de Pseudo-PSSMs, se puede observar que los sitios de unión a Calcio y Magnesio mostraron ser similares en los primeros tres aminoácidos que rodean al metal, donde ambos metales prefieren interactuar con ácido aspártico y ácido glutámico. Sólo presentaron pequeñas diferencias en el cuarto aminoácido más cercano (Ver figura 7 y 8). Esta similitud puede ser consecuencia de la preferencia de ambos metales por interactuar con los átomos de oxígeno (Dudev y Lim, 2003) presentes en la cadenas laterales de ácido aspártico y ácido glutámico. Probablemente esta similitud sea uno de los principales obstáculos en la discriminación de ambos metales en estudios anteriores (Bordner, 2008). Por otro lado el Zinc presentó diferencias con los otros metales, debido a que este prefiere interacciones con átomos de nitrógeno y azufre (Dudev y Lim, 2003), presentes en las cadenas laterales de histidina y cisteína (Ver figura 9). Estas tendencias en las frecuencias de aminoácidos alrededor de los metales corroboran los resultados obtenidos por estudios anteriores (Brylinski y Skolnick, 2011). Además cabe mencionar que la tendencia de aminoácidos polares y polares cargados en las posiciones más cercanas al metal, ya había sido descrita anteriormente por Anfinsen y col., 1991).

Las distancias entre el metal y el átomo del residuo más cercano (primera interacción) es un atributo documentado con anterioridad en (Harding, 2006), lo cual

valida los resultados obtenidos en el presente estudio. Estas distancias muestran que de los tres metales, el Zn forma las interacciones más cortas con los átomos de las cadenas laterales de los aminoácidos. Anteriormente se ha descrito que el Zn tiende a formar interacciones más fuertes, cortas y de baja movilidad con moléculas orgánicas (Crichton, 2008).

En la estimación de atributos fisicoquímicos se pudo observar que el cuarto residuo más cercano al metal parece ser el más hidrofóbico de los cuatro primeros. Además, estos iones metálicos prefieren aminoácidos más hidrofílicos en las primeras posiciones (Ver tabla 6). En la composición atómica de la capa 3 se pueden observar diferencias para estos tres metales, en donde para Ca y Mg predomina la presencia de átomos de oxígeno, de las cadenas laterales de ácido aspártico y ácido glutámico, y para el Zn no existe tal preferencia. De hecho se puede apreciar que los porcentajes de los 4 átomos son similares (Ver tabla 7). Además, se detectó que los tres metales privilegian los aminoácidos polares sin carga en el sitio completo, pero si se considera sólo los polares negativos y positivos hay una clara tendencia del Ca y el Mg por los cargados negativos, en cambio el Zn prefiere los cargados positivos.

Relacionado al desempeño obtenido por los modelos de predicción basados en SVM, se puede observar que altos valores de exactitud obtuvo la discriminación entre sitios de unión a Ca, Mg y Zn. Considerando los tipos de información de manera individual, la información fisicoquímica obtuvo la mayor exactitud, dejando en segundo lugar la información geométrica y en tercer lugar la información evolutiva (Ver figura 10). Esto sugiere que modelos basados en información fisicoquímica son los mejores para predecir sitios de unión a metales. Cuando los tipos de información fueron combinados se puede observar un incremento de exactitud.

La combinación de diversos tipos de información no sólo favorece la exactitud global de modelo, sino que además las tasas de verdaderos positivos y falsos positivos por cada clase también fueron incrementadas. La información evolutiva puede predecir los sitios de unión a Zn, pero para Ca y más aún para Mg, las tasas

de verdaderos positivos son más bajas que las otras tasas, 0,67 y 0,48 respectivamente. Las tasas de falsos positivos, cuando la información evolutiva es utilizada, sugieren que probablemente una cantidad importante de sitios de unión Mg está siendo clasificada como sitios de unión a Ca. El mismo problema, pero en menor medida ocurre para el modelo basado en información geométrica. Sin embargo, cuando estos 2 tipos de información son combinados se amortigua esta deficiencia en la discriminación. El modelo que integra los tres tipos de información mejora el desempeño en la predicción de sitios de unión a Ca y Mg. Además, mejora el equilibrio entre las tasas de verdaderos positivos y falsos positivos los diferentes sitios de unión a metales.

Para enfrentar un posible problema de alta dimensionalidad en modelo que integraba los tres tipos de información se aplicó una estrategia de 2 etapas. Se descartaron 25 atributos en total, 12 mediante coeficientes de correlación de Pearson y 13 eliminando atributos que generaban ruido al modelo. De los 13 atributos eliminados en la segunda etapa 8 están relacionados en alguna medida con el cuarto aminoácido más cercano al metal, lo cual puede indicar que este aminoácido tiene una menor importancia a la hora de discriminar entre sitios de unión a metales, en comparación a los otros 3 más cercanos. El segundo atributo más ruidoso es el Score Zn-PSSM, y con la eliminación de este atributo, sólo 1 atributo de tipo evolutivo fue conservado en el modelo. Esto permite sugerir que el aporte predictivo de la información evolutiva es menor al de los otros dos tipos de información.

En relación a la comparación de los modelos de aprendizaje supervisado, SVM mostró exactitudes superiores a DT y NBC. Pero los resultados de LR fueron muy similares a SVM, lo cual es inesperado. Hasta ahora no hay métodos de predicción ni discriminación de sitios de unión a metales basados en regresión logística. De hecho, esta técnica no es usualmente utilizada en bioinformática. Esta observación abre una nueva oportunidad de estudio que no se ha reportado anteriormente.

CONCLUSIONES

A partir de la investigación realizada se pudo construir un modelo eficiente de predicción y discriminación entre sitios de unión a metales en proteínas. Logrando un desempeño sobre el 90% de exactitud, superior al de los métodos anteriores. Además, se concluye que:

El principal obstáculo que trabajos anteriores tuvieron a la hora de discriminar entre sitios de unión a Calcio y Magnesio se debe probablemente a que ambos prefieren interaccionar con ácido glutámico y ácido aspártico en los aminoácidos más cercanos al metal. Por otro lado, el Zinc es un metal que prefiere interaccionar con Histidina en las primeras 2 posiciones y con Cisteína en las 3ra y 4ta posición, lo cual lo diferencia en gran medida con el Calcio y Magnesio.

La exactitud global del modelo de discriminación entre sitios de unión a metales, aumenta cuando diversos tipos de información son combinados, e individualmente la información fisicoquímica entrega los mejores resultados. Por otro lado, las tasas de verdaderos positivos también se ven favorecidas por la combinación de diversos tipos de información, y las tasas de falsos positivos se hacen muy pequeñas cuando se integran los tres tipos de información, es decir se reduce en gran medida la cantidad de sitios mal clasificados.

La reducción de dimensionalidad realizada mediante eliminación de atributos correlacionados y eliminación de atributos que producían ruido al modelo incrementa el desempeño en el modelo basado en información evolutiva, geométrica y fisicoquímica.

El modelo de discriminación entre metales basado en Máquina de Vectores de

Soporte supera en exactitud a los modelos basados en Árboles de Decisión y Clasificador Bayesiano Ingenuo. Por otro lado, Regresión Logística muestra similar desempeño a Máquinas de Vectores de soporte, y por lo tanto evidencia ser una interesante propuesta para futuros modelos de predicción de sitios de unión a metales.

REFERENCIAS

- Anfinsen CB, Edsall JT, Richards FM, Eisenberg DS (Eds.). *Advances in Protein Chemistry: Metalloproteins Volume 42*. Academic Press Inc. 1991
- Bagley SC, Altman RB. Characterizing the microenvironment surrounding protein sites. *Protein science*. 1995; 4(4):622–35.
- Barnham KJ, Bush AI. Metals in Alzheimer's and Parkinson's diseases. *Current opinion in chemical biology*. 2008; 12(2):222–8.
- Berman HM, Westbrook J, Feng, Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. *Nucleic acids research*. 2000; 28(1):235–42.
- Bertini I, Cavallaro G. Metals in the “omics” world: copper homeostasis and cytochrome c oxidase assembly in a new light. *Journal of biological inorganic chemistry*. 2008; 13(1):3–14.
- Binet MB, Ma R, McLeod CW, Poole RK. Detection and characterization of zinc- and cadmium-binding proteins in *Escherichia coli* by gel electrophoresis and laser ablation-inductively coupled plasma-mass spectrometry. *Analytical biochemistry*. 2003; 318(1):30–8.
- Bishop CM. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer. 2007.
- Bordner AJ. Predicting small ligand binding sites in proteins using backbone structure. *Bioinformatics*. 2008; 24(24):2865–71.
- Bouton CM, Pevsner J. Effects of lead on gene expression. *Neurotoxicology*. 2000; 21(6):1045–55.

- Brylinski M, Skolnick J. FINDSITE-metal: integrating evolutionary information and machine learning for structure-based metal-binding site prediction at the proteome level. *Proteins*. 2011; 79(3):735–51.
- Capra J, Laskowski R, Thornton JM, Singh M, Funkhouser T. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLoS computational biology*. 2009; 5(12):1-18
- Carafoli E. Calcium signaling: a tale for all seasons. *Proceedings of the National Academy of Sciences of the United States of America*. 2002; 99(3):1115–22.
- Chang CC, Lin CJ. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011; 2(3):27.
- Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. *The EMBO journal*. 1986; 5(4):823–6
- Cilia E, Passerini A. Automatic prediction of catalytic residues by modeling residue structural neighborhood. *BMC bioinformatics*. 2010; 11:115.
- Cobbett C, Goldsbrough P. Phytochelatins and metallothioneins: roles in heavy metal detoxification and homeostasis. *Annual review of plant biology*. 2002; 53:159–82.
- Cox EH, McLendon GL. Zinc-dependent protein folding. *Current opinion in chemical biology*. 2000; 4(2):162–5.
- Crichton R. *Biological Inorganic Chemistry: An Introduction*. Elsevier Science. 2008.
- Degtyarenko K. Bioinorganic motifs: towards functional classification of metalloproteins. *Bioinformatics*. 2000; 16(10):851–64.
- Dokmanić I, Sikić M, Tomić S. Metals in proteins: correlation between the metal-ion type, coordination number and the amino-acid residues involved in the

- coordination. *Acta crystallographica. Section D, Biological crystallography*. 2008; 64(3):257–63.
- Dudev T, Lim C. Principles governing Mg, Ca, and Zn binding and selectivity in proteins. *Chemical reviews*. 2003; 103(3):773–88.
- Ebert J, Altman R. Robust recognition of zinc binding sites in proteins. *Protein Science*. 2008; 17(1):54–65.
- Feng M, Patel D, Dervan JJ, Ceska T, Suck D, Haq I, Sayers JR. Roles of divalent metal ions in flap endonuclease-substrate interactions. *Nature structural & molecular biology*. 2004; 11(5):450–6.
- Finkelstein J. Metalloproteins. *Nature*. 2009; 460(7257):813.
- Goyal K, Mande SC. Exploiting 3D structural templates for detection of metal-binding sites in protein structures. *Proteins*. 2008; 70(4):1206–18.
- Guerois R, Nielsen JE, Serrano L. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of molecular biology*. 2002; 320(2):369–87.
- Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*. 2009; 11(1):10
- Harding MM. Small revisions to predicted distances around metal sites in proteins. *Acta Crystallographica Section D Biological Crystallography*. 2006; 62(6):678–682.
- Harris ED. Cellular copper transport and metabolism. *Annual review of nutrition*. 2000; 20:291–310.

- Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics). Springer. 2009.
- Hessa T, Kim H, Bihlmaier K, Lundin C, Boekel J, Andersson H, Nilsson I. Recognition of transmembrane helices by the endoplasmic reticulum translocon. *Nature*. 2005; 433(7024):377–81.
- Holm RH, Kennepohl P, Solomon EI. Structural and Functional Aspects of Metal Sites in Biology. *Chemical Reviews*. 1996; 96(7):2239–2314.
- Hsu C, Chang C, Lin C. A practical guide to support vector classification. [*en linea*], 2003. Disponible en la Web: <http://www.cs.manchester.ac.uk/pgt/2011/COMP61011/materials/practicalsvmguide.pdf>
- Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. *Journal of molecular graphics*. 1996; 14(1):33–8.
- Jensen MR, Petersen G, Lauritzen C, Pedersen J, Led JJ. Metal Binding Sites in Proteins: Identification and Characterization by Paramagnetic NMR Relaxation. *Biochemistry*. 2005; 44(33):11014–11023.
- Kumar S, Nussinov R. Close Range Electrostatic Interactions in Proteins Close-Range Electrostatic Interactions in Proteins. *ChemBioChem*. 2002; 3(7):604–17.
- Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*. 1982; 157(1):105–32.
- Larrañaga P, Calvo B, Santana R, y col. Machine learning in bioinformatics. *Briefings in Bioinformatics*. 2006; 7(1):86–112.
- Lieu PT, Heiskala M, Peterson PA, Yang Y. The roles of iron in health and disease. *Molecular aspects of medicine*. 2001; 22(1-2):1–87.

- Lin CT, Lin KL, Yang CH, Chung IF, Huang CD, Yang YS. Protein metal binding residue prediction based on neural networks. *International journal of neural systems*. 2005; 15(1-2):71–84.
- Lin HH, Han LY, Zhang HL, Zheng CJ, Xie B, Cao ZW, Chen YZ. Prediction of the functional class of metal-binding proteins from sequence derived physicochemical properties by support vector machine approach. *BMC Bioinformatics*. 2006; 7(5):13.
- Marsland S. *Machine Learning: An Algorithmic Perspective* (Chapman & Hall/Crc Machine Learning & Pattern Recognition). Chapman and Hall/CRC. 2009.
- Mitchell TM. *Machine Learning*. McGraw-Hill. 1997.
- Nassif H, Al-Ali H, Khuri S, Keirouz W. Prediction of protein-glucose binding sites using support vector machines. *Proteins*. 2009; 77(1):121–32.
- Papoyan A, Kochian LV. Identification of *Thlaspi caerulescens* genes that may be involved in heavy metal hyperaccumulation and tolerance. Characterization of a novel heavy metal transporting ATPase. *Plant physiology*. 2004; 136(3):3814–23.
- Passerini A, Andreini C, Menchetti S, Rosato A, Frasconi P. Predicting zinc binding at the proteome level. *BMC Bioinformatics*. 2007; 8:39.
- Passerini A, Punta M, Ceroni A, Rost B, Frasconi P. Identifying cysteines and histidines in transition-metal-binding sites using support vector machines and neural networks. *Proteins*. 2006; 65(2):305–316.
- Reed GH, Poyner RR. Mn^{2+} as a probe of divalent metal ion binding and function in enzymes and other proteins. *Metal ions in biological systems*. 2000; 37:183–207.
- Rice P, Longden I, Bleasby A. EMBOSS: the European Molecular Biology Open Software Suite. *Trends in genetics : TIG*. 2000; 16(6):276–7.

- Schymkowitz JWH, Rousseau F, Martins IC, Ferkinghoff-Borg J, Stricher F, Serrano L. Prediction of water and metal binding sites and their affinities by using the Fold-X force field. *Proceedings of the National Academy of Sciences of the United States of America*. 2005; 102(29):10147–52.
- Shi W, Chance MR. Metallomics and metalloproteomics. *Cellular and molecular life sciences : CMLS*. 2008; 65(19):3040–8.
- Shu N, Zhou T, Hovmöller S. Prediction of zinc-binding sites in proteins from sequence. *Bioinformatics*. 2008; 24(6):775–82
- Sodhi JS, Bryson K, McGuffin LJ, Ward JJ, Wernisch L, Jones DT. Predicting metal-binding site residues in low-resolution structural models. *Journal of molecular biology*. 2004; 342(1):307–20.
- Szpunar J. Advances in analytical methodology for bioinorganic speciation analysis: metallomics, metalloproteomics and heteroatom-tagged proteomics and metabolomics. *The Analyst*. 2005; 130(4):442–65.
- Vapnik VN. *The Nature of Statistical Learning Theory*. Springer. 1998.
- Wang G, Dunbrack RL. PISCES: a protein sequence culling server. *Bioinformatics*. 2003; 19(12):1589–1591.
- Wei L, Huang ES, Altman RB. Are predicted structures good enough to preserve functional sites? *Structure*. 1999; 7(6):643–50.
- Wimley WC, White SH. Experimentally determined hydrophobicity scale for proteins at membrane interfaces. *Nature structural biology*. 1996; 3(10):842–8.
- Zhang Y, Rajapakse JC. *Machine Learning in Bioinformatics*. Wiley. 2008.

Zhou T, Hamer DH, Hendrickson WA, Sattentau QJ, Kwong PD. Interfacial metal and antibody recognition. *Proceedings of the National Academy of Sciences of the United States of America*. 2005; 102(41):14575–80.

ANEXOS

Anexo 1.1

Script en Perl para identificar las cadenas-M que unen Zn, Mg o Ca desde
las estructuras de metaloproteínas descargadas

```
#!/usr/bin/perl
use warnings;
use strict;

my $n_pdb = 0;
my $n_chains = 0;
my $n_Mchains = 0;
my $n_Znchains = 0;
my $n_Mgchains = 0;
my $n_Cachains = 0;

open(IN, "PDBs_Zn_Mg_Ca.txt");
open(OUT, ">>Mchains_Zn_Mg_Ca.txt");

while(<IN>){

    my @id = split(".p", $_);
    my $n_comas = 0;
    my $n_chains_temp = 0;
    my $metal = '';
    my $chain = '';
    my @Mchains = ();
    my @Cachains = ();
    my @Mgchains = ();
    my @Znchains = ();
    my $Cacount = 0;
    my $Mgcount = 0;
    my $Zncount = 0;

    open(PDB, "PDBs_Zn_Mg_Ca_18_08_2011/$_");

    while(<PDB>){

        if (/^COMPND/ && (substr($_, 11, 5) =~ "CHAIN")){
            $n_comas = ($_ =~ tr/,//);
            $n_chains_temp = $n_chains_temp + ($n_comas + 1);
        }

        if (/^HETATM/){
            $metal = substr($_, 18, 2);

            if($metal =~ "CA"){
                $chain = substr($_, 21, 1);
                $Cachains[$Cacount] = $chain;
                $Cacount++;
            }
            if($metal =~ "MG"){
```

```

        $chain = substr($_,21,1);
        $Mgchains[$Mgcount] = $chain;
        $Mgcount++;
    }
    if($metal =~ "ZN" ){
        $chain = substr($_,21,1);
        $Znchains[$Zncount] = $chain;
        $Zncount++;
    }
}
}
close(PDB);

my %hash1 = map { $_ => 1 } @Cachains;
my @Cachains_nr = keys %hash1;

my %hash2 = map { $_ => 1 } @Mgchains;
my @Mgchains_nr = keys %hash2;

my %hash3 = map { $_ => 1 } @Znchains;
my @Znchains_nr = keys %hash3;

@Mchains = (@Cachains_nr,@Mgchains_nr,@Znchains_nr);

my %hash4 = map { $_ => 1 } @Mchains;
my @Mchains_nr = keys %hash4;

my $n_Cachains_temp = @Cachains_nr;
my $n_Mgchains_temp = @Mgchains_nr;
my $n_Znchains_temp = @Znchains_nr;
my $n_Mchains_temp = @Mchains_nr;

for(my $j=0;$j<$n_Mchains_temp;$j++){
    print OUT "$id[0]$Mchains_nr[$j]\n"
}

$n_chains = $n_chains + $n_chains_temp;
$n_Cachains = $n_Cachains + $n_Cachains_temp;
$n_Mgchains = $n_Mgchains + $n_Mgchains_temp;
$n_Znchains = $n_Znchains + $n_Znchains_temp;
$n_Mchains = $n_Mchains + $n_Mchains_temp;
$n_pdb++};

print "PDB: $n_pdb\nChains: $n_chains\nMchains: $n_Mchains\nCachains:
$n_Cachains\nMgchains: $n_Mgchains\nZnchains: $n_Znchains\n\n";

```

Anexo 1.2

Script en Tcl para extraer los sitios de unión metales presentes en las
cadenas-M no redundantes

```

proc extract_sites {metal metaldir} {
    set IN [open Mchains_NR.txt]
    while {-1 != [gets $IN line]} {
        set id_chain [split $line "-"]
    }
}

```

```

set id [lindex $id_chain 0]
set chain [lindex $id_chain 1]

mol new "PDBs_NR/$id.pdb"
set ions [atomselect top "chain $chain and resname $metal"]
set number [$ions num]
set indexs [$ions get index]

set no_index {}
if { $number > 1 } {
  for {set i 0} {$i<$number} {incr i} {
    for {set j 0} {$j<$number} {incr j} {
      if {$i < $j} {
        set bond [measure bond [list [lindex $indexs $i] [lindex
$indexs $j]]]
        if {$bond < 14} {
          lappend no_index [lindex $indexs $i] [lindex $indexs $j]
        }
      }
    }
  }
}

set no_index_nr [lsort -unique $no_index]

set large [llength $no_index_nr]

for {set i 0} {$i<$large} {incr i} {
  set idx [lsearch $indexs [lindex $no_index_nr $i]]
  set indexs [lreplace $indexs $idx $idx]
}

set large2 [llength $indexs]

if {$large2 != 0} {
  for {set i 0} {$i<$large2} {incr i} {
    set temp [lindex $indexs $i]
    set site_temp [atomselect top "name CA and backbone and same
residue as within 7 of index $temp"]
    set aa_num [$site_temp num]
    set site_temp2 [atomselect top "name CA and backbone and same
residue as within 3 of index $temp"]
    set aa_num2 [$site_temp2 num]
    if {$aa_num >= 4 && $aa_num2 >= 1} {
      set site [atomselect top "((resname ALA ARG ASN ASP CYS GLN GLU
GLY HIS ILE LEU LYS MET PHE PRO SER THR TRP TYR VAL) or index $temp) and
same residue as within 7 of index $temp"]
      $site writepdb $metaldir/$id.$temp.$chain.pdb
    }
  }
}

mol delete all
}

file mkdir Ca_sites

```

```

file mkdir Mg_sites
file mkdir Zn_sites

extract_sites CA Ca_sites
extract_sites MG Mg_sites
extract_sites ZN Zn_sites

exit

```

Anexo 1.3

```

# Script en Perl para identificar sitios de unión a metales con más de una
# conformación

#!/usr/bin/perl
use warnings;
use strict;

sub make_list {
    my $filein = $_[0]."_sites.txt";
    my $dirin = $_[0]."_sites";
    my $fileout = $_[0]."_2conformation.txt";

    open(IN, "$filein");
    open(OUT, ">>$fileout");
    my $count = 0;
    while(<IN>){
        open(PDB, "$dirin/$_");
        my $id = $_;
        chomp($id);
        my $temp = '_';
        my $aux = '';
        while(<PDB>){
            if(/^ATOM/){
                my $letter = substr($_,16,1);
                if($letter eq 'B' && $aux !~ $temp){
                    $aux = "$id-$letter";
                    print OUT "$id\n";
                    $temp = $aux;
                    $count++;
                }
            }
        }
        print "$_[0]: $count\n";
    }

    &make_list("Ca");
    &make_list("Mg");
    &make_list("Zn");
}

```

Anexo 2.1

```

# Script en Perl para obtener la secuencia en formato fasta de los 4
# aminoácidos más cercanos al metal para cada sitio de unión a metal

```

```

#!/usr/bin/perl
use warnings;
use strict;

sub traduction {
    my $in = $_[0];
    my $out = '';
    if($in eq "ALA") {$out="A";}
    if($in eq "CYS") {$out="C";}
    if($in eq "ASP") {$out="D";}
    if($in eq "GLU") {$out="E";}
    if($in eq "PHE") {$out="F";}
    if($in eq "GLY") {$out="G";}
    if($in eq "HIS") {$out="H";}
    if($in eq "ILE") {$out="I";}
    if($in eq "LYS") {$out="K";}
    if($in eq "LEU") {$out="L";}
    if($in eq "MET") {$out="M";}
    if($in eq "ASN") {$out="N";}
    if($in eq "PRO") {$out="P";}
    if($in eq "GLN") {$out="Q";}
    if($in eq "ARG") {$out="R";}
    if($in eq "SER") {$out="S";}
    if($in eq "THR") {$out="T";}
    if($in eq "VAL") {$out="V";}
    if($in eq "TRP") {$out="W";}
    if($in eq "TYR") {$out="Y";}
    return $out;
}

sub write_fasta {
    open(OUT,">>$_[2]");
    open(IN,"$_[1]");
    while(<IN>){
        open(PDB,"$_[0]/$_");
        my $site_ID = $_;
        chomp($site_ID);
        my $next = -1;
        my $no = 1;
        my @array_d = ();
        my @array_aa = ();
        my @array_atoms = ();
        my $distance = 50;
        my $McoorX = '';
        my $McoorY = '';
        my $McoorZ = '';
        my $temp = '';

        while(<PDB>){
            if(/^ATOM/ && substr($_,17,2) eq $_[3] && $no == 1){
                $McoorX = substr($_,30,8);
                $McoorY = substr($_,38,8);
                $McoorZ = substr($_,46,8);
                seek(PDB,0,0);
                $no = 0;
            }
        }
    }
}

```

```

if(/^ATOM/ && $no == 0){
    my $aa3 = substr($_,17,3);
    my $n = substr($_,23,3);
    my $aa3_n = $aa3.$n;
    my $atom = substr($_,13,3);

    if($atom eq 'N '){
        $temp = $aa3_n;
        $distance = 50;
        $next++;
    }

    if($aa3 ne $_[3] && $aa3_n eq $temp){
        my $coorX = substr($_,30,8);
        my $coorY = substr($_,38,8);
        my $coorZ = substr($_,46,8);
        my $d1 = $McoorX - $coorX;
        my $d2 = $McoorY - $coorY;
        my $d3 = $McoorZ - $coorZ;
        my $d4 = $d1*$d1 + $d2*$d2 + $d3*$d3;
        my $d5 = sqrt($d4);
        if($d5<$distance){
            $distance = $d5;
        }
        $array_d[$next] = $distance;
        $array_aa[$next] = $aa3;
        $array_atoms[$next] = $atom;
        $temp = $aa3_n;
    }
}

my @sorted_array_d = sort(@array_d);
print OUT ">$site_ID\n";
my $large = @array_aa;
for(my $i=0;$i<4;$i++){
    for(my $j=0;$j<$large;$j++){
        if($sorted_array_d[$i] == $array_d[$j]){
            print OUT &traduction($array_aa[$j]);
        }
    }
}
print OUT "\n\n";
}

&write_fasta('Ca_sites','Ca_sites.txt','Ca_sites.fastas','CA');
&write_fasta('Mg_sites','Mg_sites.txt','Mg_sites.fastas','MG');
&write_fasta('Zn_sites','Zn_sites.txt','Zn_sites.fastas','ZN');

```

Anexo 2.2

```

# Script en Perl para identificar las coordenadas del ion metálico y los
# átomos de las cadenas laterales y carbonos alfa de los 4 aminoácidos más
# cercanos al metal

```

```

#!/usr/bin/perl
use warnings;
use strict;

sub coordinates{

    my $metal = $_[0];
    my $file_in = $metal."_sites.txt";
    my $metal2 = $metal;
    $metal2 =~ tr/[a-z]/[A-Z]/;

    open(IN,$file_in);
    my $folder = $metal."_sites";

    open(OUT,">>coord$metal.txt");

    while(<IN>){
        open(PDB,"$folder/$_");
        my $site_ID = $_;
        chomp($site_ID);

        my $next_aa = -1;
        my $skip_metal = 1;

        my $McoorX = '';
        my $McoorY = '';
        my $McoorZ = '';
        my @array_d = ();
        my @array_aa = ();
        my @array_atoms = ();
        my @array_n = ();
        my @coorX = ();
        my @coorY = ();
        my @coorZ = ();
        my @CcoorX = ();
        my @CcoorY = ();
        my @CcoorZ = ();
        my $distance = 50;
        my $temp = '';

        while(<PDB>){
            if(/^ATOM/ && substr($_,17,2) eq $metal2 && $skip_metal == 1){
                $McoorX = substr($_,30,8);
                $McoorY = substr($_,38,8);
                $McoorZ = substr($_,46,8);
                seek(PDB,0,0);
                $skip_metal = 0;
            }

            if(/^ATOM/ && $skip_metal == 0){
                my $aa3 = substr($_,17,3);
                my $n = substr($_,23,3);
                my $aa3_n = $aa3.$n;
                my $atom = substr($_,13,3);

                if($atom eq 'N '){

```

```

    $temp = $aa3_n;
    $distance = 50;
    $next_aa++;
}

if($atom eq 'CA ' && $aa3 !~ 'CA '){
    my $CcoorX = substr($_,30,8);
    my $CcoorY = substr($_,38,8);
    my $CcoorZ = substr($_,46,8);
    $CcoorX[$next_aa] = $CcoorX;
    $CcoorY[$next_aa] = $CcoorY;
    $CcoorZ[$next_aa] = $CcoorZ;
}

if($atom ne 'CA ' && $aa3 !~ $metal2 && $aa3_n eq $temp){
    my $coorX = substr($_,30,8);
    my $coorY = substr($_,38,8);
    my $coorZ = substr($_,46,8);
    my $d1 = $McoorX - $coorX;
    my $d2 = $McoorY - $coorY;
    my $d3 = $McoorZ - $coorZ;
    my $d4 = $d1*$d1 + $d2*$d2 + $d3*$d3;
    my $d5 = sqrt($d4);
    if($d5<$distance){
        $distance = $d5;
        $array_d[$next_aa] = $distance;
        $array_aa[$next_aa] = $aa3;
        $array_atoms[$next_aa] = $atom;
        $array_n[$next_aa] = $n;
        $coorX[$next_aa] = $coorX;
        $coorY[$next_aa] = $coorY;
        $coorZ[$next_aa] = $coorZ;
        $temp = $aa3_n;
    }
}
}
}

my @positions = ();
my @sorted_array_d = sort(@array_d);
for(my $i=0;$i<4;$i++){
    for(my $j=0;$j<@array_d;$j++){
        if($sorted_array_d[$i]==$array_d[$j]){
            $positions[$i] = $j;
        }
    }
}

print OUT "$site_ID,$McoorX,$McoorY,$McoorZ";
for(my $i=0;$i<4;$i++){
    print OUT
    ",$coorX[$positions[$i]],$coorY[$positions[$i]],$coorZ[$positions[$i]]";
    print OUT
    ",$CcoorX[$positions[$i]],$CcoorY[$positions[$i]],$CcoorZ[$positions[$i]]";
}

```



```

        print OUT "\n";
    }
}

&coordinates('Zn');
&coordinates('Mg');
&coordinates('Ca');

```

Anexo 2.3

```

# Script en Perl para estimar los atributos geométricos basados en
# distancias y ángulos

#!/usr/bin/perl
use warnings;
use strict;

sub cal_dist {
    my $d1 = $_[0] - $_[3];
    my $d2 = $_[1] - $_[4];
    my $d3 = $_[2] - $_[5];
    my $d4 = $d1*$d1 + $d2*$d2 + $d3*$d3;
    my $d5 = sqrt($d4);
    return $d5;
}

sub write_file {

    open(OUT1,">>dist_a_c_$_[0]");
    print OUT1 "ID,d_m_a1,d_m_c1,d_m_a2,d_m_c2,d_m_a3,d_m_c3,d_m_a4,d_m_c4";

    open(OUT2,">>dist_a_a_$_[0]");
    print OUT2 "ID,d_a1_a2,d_a1_a3,d_a1_a4,d_a2_a3,d_a2_a4,d_a3_a4";

    open(OUT3,">>dist_c_c_$_[0]");
    print OUT3 "ID,d_c1_c2,d_c1_c3,d_c1_c4,d_c2_c3,d_c2_c4,d_c3_c4";

    open(OUT4,">>ang_a_m_a_$_[0]");
    print OUT4
"ID,an_a1_m_a2,an_a1_m_a3,an_a1_m_a4,an_a2_m_a3,an_a2_m_a4,an_a3_m_a4";

    open(OUT5,">>ang_c_m_c_$_[0]");
    print OUT5
"ID,an_c1_m_c2,an_c1_m_c3,an_c1_m_c4,an_c2_m_c3,an_c2_m_c4,an_c3_m_c4";

    open(OUT6,">>ang_c_a_m_$_[0]");
    print OUT6 "ID,an_c1_a1_m,an_c2_a2_m,an_c3_a3_m,an_c4_a4_m";

    open(IN,"coor$_[0].txt");
    while(<IN>){
        my @coor = split ("",$_);
        print OUT1 "\n$coor[0]";
        for(my $i=0;$i<24;$i=$i+6){
            my $dist1 =
&cal_dist($coor[1],$coor[2],$coor[3],$coor[$i+4],$coor[$i+4+1],$coor[$i+4+2
]);

```

```

        print OUT1 ", $dist1";
        my $dist2 =
&cal_dist($coor[1], $coor[2], $coor[3], $coor[$i+7], $coor[$i+7+1], $coor[$i+7+2
]);
        print OUT1 ", $dist2";
    }
    print OUT2 "\n$coor[0]";
    for(my $i=0; $i<24; $i=$i+6) {
        for(my $j=0; $j<24; $j=$j+6) {
            if($i<$j) {
                my $dist1 =
&cal_dist($coor[$i+4], $coor[$i+4+1], $coor[$i+4+2], $coor[$j+4], $coor[$j+4+1]
, $coor[$j+4+2]);
                print OUT2 ", $dist1";
            }
        }
    }
    print OUT3 "\n$coor[0]";
    for(my $i=0; $i<24; $i=$i+6) {
        for(my $j=0; $j<24; $j=$j+6) {
            if($i<$j) {
                my $dist1 =
&cal_dist($coor[$i+7], $coor[$i+7+1], $coor[$i+7+2], $coor[$j+7], $coor[$j+7+1]
, $coor[$j+7+2]);
                print OUT3 ", $dist1";
            }
        }
    }
    print OUT4 "\n$coor[0]";
    for(my $i=0; $i<24; $i=$i+6) {
        for(my $j=0; $j<24; $j=$j+6) {
            if($i<$j) {
                my $angle = `python ang.py $coor[$i+4] $coor[$i+4+1]
$coor[$i+4+2] $coor[1] $coor[2] $coor[3] $coor[$j+4] $coor[$j+4+1]
$coor[$j+4+2]`;
                chomp($angle);
                if ($angle > 180) {
                    my $angle2 = (360 - $angle);
                    print OUT4 ", $angle2";
                }
                if ($angle <= 180) {
                    print OUT4 ", $angle";
                }
            }
        }
    }
    print OUT5 "\n$coor[0]";
    for(my $i=0; $i<24; $i=$i+6) {
        for(my $j=0; $j<24; $j=$j+6) {
            if($i<$j) {
                my $angle = `python ang.py $coor[$i+7] $coor[$i+7+1]
$coor[$i+7+2] $coor[1] $coor[2] $coor[3] $coor[$j+7] $coor[$j+7+1]
$coor[$j+7+2]`;
                chomp($angle);
                #print "$angle\n";
                if ($angle > 180) {

```

```

        my $angle2 = (360 - $angle);
        print OUT5 ",$angle2";
    }
    if ($angle <= 180) {
        print OUT5 ",$angle";
    }
}
}
}
print OUT6 "\n$coor[0]";
for(my $i=0;$i<24;$i=$i+6){
    my $angle = `python ang.py $coor[1] $coor[2] $coor[3] $coor[$i+4]
$coor[$i+4+1] $coor[$i+4+2] $coor[$i+7] $coor[$i+7+1] $coor[$i+7+2]`;
    chomp($angle);
    if ($angle > 180) {
        my $angle2 = (360 - $angle);
        print OUT6 ",$angle2";
    }
    if ($angle <= 180) {
        print OUT6 ",$angle";
    }
}
}
}

&write_file('Zn');
&write_file('Mg');
&write_file('Ca');

```

Anexo 2.4

Script en *Python* calcular un ángulo en base a la fórmula del producto
escalar

```

import sys
import math

v1_1 = float(sys.argv[1])
v1_2 = float(sys.argv[2])
v1_3 = float(sys.argv[3])
m_1 = float(sys.argv[4])
m_2 = float(sys.argv[5])
m_3 = float(sys.argv[6])
v2_1 = float(sys.argv[7])
v2_2 = float(sys.argv[8])
v2_3 = float(sys.argv[9])

v1= [v1_1,v1_2,v1_3]
m= [m_1,m_2,m_3]
v2= [v2_1,v2_2,v2_3]

t1 = [0,0,0]
t2 = [0,0,0]

o = [0,0,0]

```

```

for i in range(3):
    t1[i] = v1[i] - m[i]

for i in range(3):
    t2[i] = v2[i] - m[i]

a1=sum((a*b) for a, b in zip(t1, t2))

x1=sum(((a-b)*(a-b)) for a, b in zip(t1, o))
x2=sum(((a-b)*(a-b)) for a, b in zip(t2, o))

b1=math.sqrt(x1)
b2=math.sqrt(x2)

y1=(a1/(b1 * b2))

a3=math.acos(y1)*57.2957795
print a3

```

Anexo 2.5

Script en Perl para estimar los atributos fisicoquímicos basados en
hidrofobicidades

```

#!/usr/bin/perl
use warnings;
use strict;

sub cal_hidro {
    my $hidros = '';
    if ($_[0] eq 'I') {$hidros = "4.5,0.31,-0.60"; return $hidros;}
    if ($_[0] eq 'V') {$hidros = "4.2,-0.07,-0.31"; return $hidros;}
    if ($_[0] eq 'L') {$hidros = "3.8,0.56,-0.55"; return $hidros;}
    if ($_[0] eq 'F') {$hidros = "2.8,1.13,-0.32"; return $hidros;}
    if ($_[0] eq 'C') {$hidros = "2.5,0.24,-0.13"; return $hidros;}
    if ($_[0] eq 'M') {$hidros = "1.9,0.23,-0.10"; return $hidros;}
    if ($_[0] eq 'A') {$hidros = "1.8,-0.17,0.11"; return $hidros;}
    if ($_[0] eq 'G') {$hidros = "-0.4,-0.01,0.74"; return $hidros;}
    if ($_[0] eq 'T') {$hidros = "-0.7,-0.14,0.52"; return $hidros;}
    if ($_[0] eq 'S') {$hidros = "-0.8,-0.13,0.84"; return $hidros;}
    if ($_[0] eq 'W') {$hidros = "-0.9,1.85,0.30"; return $hidros;}
    if ($_[0] eq 'Y') {$hidros = "-1.3,0.94,0.68"; return $hidros;}
    if ($_[0] eq 'P') {$hidros = "-1.6,-0.45,2.23"; return $hidros;}
    if ($_[0] eq 'H') {$hidros = "-3.2,-0.96,2.06"; return $hidros;}
    if ($_[0] eq 'E') {$hidros = "-3.5,-2.02,2.68"; return $hidros;}
    if ($_[0] eq 'Q') {$hidros = "-3.5,-0.58,2.36"; return $hidros;}
    if ($_[0] eq 'D') {$hidros = "-3.5,-1.23,3.49"; return $hidros;}
    if ($_[0] eq 'N') {$hidros = "-3.5,-0.42,2.05"; return $hidros;}
    if ($_[0] eq 'K') {$hidros = "-3.9,-0.99,2.71"; return $hidros;}
    if ($_[0] eq 'R') {$hidros = "-4.5,-0.81,2.58"; return $hidros;}
}

sub write_hidros {
    open(IN,"$_[0]");
    open(OUT,">$_[1]");
    my $line=1;

```

```

while(<IN>){
  my $ini1 = ">";
  my $ini2 = "\n";

  if(/^$ini1/){
    chomp($_);
    my $largo = length($_);
    my $id = substr($_,1,$largo-2);
    print OUT "$id";
  }
  elsif(/^$ini2/){
    print OUT "\n";
  }
  else {
    my @aas = split("",$_);
    for(my $i=0;$i<4;$i++){
      my $text = &cal_hidro($aas[$i]);
      print OUT ",$text";
    }
  }
}

&write_hidros('Ca_sites.fasta','Ca_hidros');
&write_hidros('Mg_sites.fasta','Mg_hidros');
&write_hidros('Zn_sites.fasta','Zn_hidros');

```

Anexo 2.6

Script en Tcl para estimar los atributos fisicoquímicos basados en
composición atómica y conteo de átomos

```

proc atoms_shell {metal metalfile metaldir} {
  set IN [open $metalfile]
  set OUT [open $metal-capas.txt "a+"]
  while {-1 != [gets $IN line]} {
    mol new "$metaldir/$line"
    puts -nonewline $OUT "$line"
    set capa {}
    for {set i 0} {$i<7} {incr i} {
      set suma 0
      foreach element {C O N S} {
        set j [expr {$i+1}]
        set atoms [atomselect top "((exwithin $j of resname $metal) and not
within $i of resname $metal) and element $element"]
        set count [$atoms num]
        lappend capa $count
        set suma [expr {$suma+$count}]
      }
      if {$suma != 0} {
        foreach n_atom {0 1 2 3} {
          set percent [expr {[lindex $capa $n_atom] * 100 / $suma}]
          puts -nonewline $OUT ",$percent"
        }
        puts -nonewline $OUT ",$suma"
      }
    }
  }
}

```

```

    if {$suma == 0} {
        foreach n_atom {0 1 2 3} {
            puts -nonewline $OUT ",0"
        }
        puts -nonewline $OUT ",$suma"
    }

    unset capa
}
puts -nonewline $OUT "\n"
mol delete all
}
close $OUT
}

atoms_shell CA Ca_sites.txt Ca_sites
atoms_shell MG Mg_sites.txt Mg_sites
atoms_shell ZN Zn_sites.txt Zn_sites

exit

```

Anexo 2.7

Script en Tcl para estimar los atributos fisicoquímicos basados en
área accesible al solvente y composición aminoacídica

```

proc cal_sasa_compo {metaldir filein fileout} {
    set IN [open $filein]
    set OUT [open $fileout "a+"]
    while {-1 != [gets $IN line]} {
        mol new "$metaldir/$line"
        puts -nonewline $OUT "$line"

        set proteina [atomselect top "protein"]
        set m_sasa [measure sasa 1.4 $proteina]

        set no_polares [atomselect top "backbone and name CA and resname ALA
TRP ILE LEU PHE PRO MET"]
        set n_no_p [$no_polares num]

        set polares_neutros [atomselect top "backbone and name CA and resname
ASN CYS GLN SER THR TYR VAL GLY"]
        set n_p_neutros [$polares_neutros num]

        set polares_positivos [atomselect top "backbone and name CA and resname
ASP GLU"]
        set n_p_positivos [$polares_positivos num]

        #arreglar

        set polares_negativos [atomselect top "backbone and name CA and resname
HIS LYS ARG"]
        set n_p_negativos [$polares_negativos num]

        set n_total [expr {$n_no_p + $n_p_neutros + $n_p_positivos +
$n_p_negativos}]
    }
}

```

```

set porc_no_p [expr {$n_no_p * 100 / $n_total}]
set porc_p_neutros [expr {$n_p_neutros * 100 / $n_total}]
set porc_p_positivos [expr { $n_p_positivos * 100 / $n_total}]
set porc_p_negativos [expr { $n_p_negativos * 100 / $n_total}]

puts -nonewline $OUT
", $m_sasa, $porc_no_p, $porc_p_neutros, $porc_p_positivos, $porc_p_negativos"

puts -nonewline $OUT "\n"

mol delete all
}
close $OUT
}

cal_sasa_compo Ca_sites Ca_sites.txt Ca_sasa_compo
cal_sasa_compo Mg_sites Mg_sites.txt Mg_sasa_compo
cal_sasa_compo Zn_sites Zn_sites.txt Zn_sasa_compo

exit

```

Anexo 3.1

```

% Script Matlab para normalizar los atributos entre 1 y -1

for i=1:101,
t = datas(:,i);
datas_normalized(:,i) = (t - mean(t)) / max(abs(t - mean(t)));
end

save('datas_normalized','datas_normalized');

```

Anexo 3.2

```

% Script Matlab para agrupar los atributos por tipo de información y la
% combinación de ellos

```

```

evo = datas_normalized(:,[1:3]);
geo = datas_normalized(:,[4:39]);
phy = datas_normalized(:,[40:101]);

evo_geo = [evo,geo];
evo_phy = [evo,phy];
geo_phy = [geo,phy];

save('evo','evo');
save('geo','geo');
save('phy','phy');

save('evo_geo','evo_geo');
save('evo_phy','evo_phy');
save('geo_phy','geo_phy');

```

Anexo 3.3

```
% Script Matlab para encontrar los parámetros óptimos para los modelos
basados en SVM

parameters = [];
bestcv = 0;
i=0;
for log2c = 0:12,
    for log2g = -12:0,
        i=i+1;
        cv = datas_crossv(datas_normalized,class_number, 2^log2c,2^log2g);
        if (cv > bestcv),
            bestcv = cv; bestc = 2^log2c; bestg = 2^log2g;
        end
        fprintf('%g %g %g (best c=%g, g=%g, rate=%g)\n', log2c, log2g, cv, bestc,
bestg, bestcv);
        parameters(i,1) = 2^log2c;
        parameters(i,2) = 2^log2g;
        parameters(i,3) = cv;
    end
end

save('parameters','parameters');
save('bestC','bestc');
save('bestg','bestg');
```

Anexo 3.4

% Script Matlab para implementar la validación cruzada de 10 iteraciones y obtener las medidas de desempeño en por clase en cada iteración

```
function [mean_total] = datas_crossv_full(info,class_number,bestc,bestg)
k=1;
l=114;
for i=1:10,
    eval(['S' num2str(i) ' = info((k:l),:);']);
    eval(['class_' num2str(i) ' = class_number((k:l),:);']);
    if i ~= 9
        k=k+114;
        l=l+114;
    elseif i==9
        k=k+114;
        l=l+122;
    end
end

t1 = [S2;S3;S4;S5;S6;S7;S8;S9;S10];
c1 =
[class_2;class_3;class_4;class_5;class_6;class_7;class_8;class_9;class_10];

t2 = [S1;S3;S4;S5;S6;S7;S8;S9;S10];
c2 =
[class_1;class_3;class_4;class_5;class_6;class_7;class_8;class_9;class_10];

t3 = [S1;S2;S4;S5;S6;S7;S8;S9;S10];
```



```

c3 =
[class_1;class_2;class_4;class_5;class_6;class_7;class_8;class_9;class_10];

t4 = [S1;S2;S3;S5;S6;S7;S8;S9;S10];
c4 =
[class_1;class_2;class_3;class_5;class_6;class_7;class_8;class_9;class_10];

t5 = [S1;S2;S3;S4;S6;S7;S8;S9;S10];
c5 =
[class_1;class_2;class_3;class_4;class_6;class_7;class_8;class_9;class_10];

t6 = [S1;S2;S3;S4;S5;S7;S8;S9;S10];
c6 =
[class_1;class_2;class_3;class_4;class_5;class_7;class_8;class_9;class_10];

t7 = [S1;S2;S3;S4;S5;S6;S8;S9;S10];
c7 =
[class_1;class_2;class_3;class_4;class_5;class_6;class_8;class_9;class_10];

t8 = [S1;S2;S3;S4;S5;S6;S7;S9;S10];
c8 =
[class_1;class_2;class_3;class_4;class_5;class_6;class_7;class_9;class_10];

t9 = [S1;S2;S3;S4;S5;S6;S7;S8;S10];
c9 =
[class_1;class_2;class_3;class_4;class_5;class_6;class_7;class_8;class_10];

t10 = [S1;S2;S3;S4;S5;S6;S7;S8;S9];
c10 =
[class_1;class_2;class_3;class_4;class_5;class_6;class_7;class_8;class_9];

MATRIX = zeros(3,3);
TPS_CA = [];
TPS_MG = [];
TPS_ZN = [];

FPS_CA = [];
FPS_MG = [];
FPS_ZN = [];

ACCS = [];

for i=1:10,
    parameter = ['-q -c ', num2str(bestc), ' -g ', num2str(bestg)];
    eval(['model = libsvmtrain(c' num2str(i) ', t' num2str(i) ',
parameter);']);
    eval(['[outA,outB,outC] = libsvmpredict(class_' num2str(i) ', S'
num2str(i) ', model);']);
    eval(['OUTA' num2str(i) ' = outA;']);
    eval(['OUTB' num2str(i) ' = outB;']);
    eval(['OUTC' num2str(i) ' = outC;']);
    eval(['pre_confu_matrix = confusionmat(class_' num2str(i) ',OUTA'
num2str(i) ');']);
    confu_matrix = pre_confu_matrix';
    disp(confu_matrix);

```

```

MATRIX = MATRIX+confu_matrix;

%ACCS(i,:) = outB(1,:);
ACCS(i,:) =
(confu_matrix(1,1)+confu_matrix(2,2)+confu_matrix(3,3))/(confu_matrix(1,1)+
confu_matrix(2,1)+confu_matrix(3,1)+confu_matrix(1,2)+confu_matrix(2,2)+con
fu_matrix(3,2)+confu_matrix(1,3)+confu_matrix(2,3)+confu_matrix(3,3));

TPS_CA(i,:) =
confu_matrix(1,1)/(confu_matrix(1,1)+confu_matrix(2,1)+confu_matrix(3,1));
TPS_MG(i,:) =
confu_matrix(2,2)/(confu_matrix(1,2)+confu_matrix(2,2)+confu_matrix(3,2));
TPS_ZN(i,:) =
confu_matrix(3,3)/(confu_matrix(1,3)+confu_matrix(2,3)+confu_matrix(3,3));
disp(TPS_CA(i,:));
disp(TPS_MG(i,:));
disp(TPS_ZN(i,:));

FPS_CA(i,:) =
(confu_matrix(1,2)+confu_matrix(1,3))/(confu_matrix(1,2)+confu_matrix(1,3)+
confu_matrix(2,2)+confu_matrix(2,3)+confu_matrix(3,2)+confu_matrix(3,3));
FPS_MG(i,:) =
(confu_matrix(2,1)+confu_matrix(2,3))/(confu_matrix(1,1)+confu_matrix(1,3)+
confu_matrix(2,1)+confu_matrix(2,3)+confu_matrix(3,1)+confu_matrix(3,3));
FPS_ZN(i,:) =
(confu_matrix(3,1)+confu_matrix(3,2))/(confu_matrix(1,1)+confu_matrix(1,2)+
confu_matrix(2,1)+confu_matrix(2,2)+confu_matrix(3,1)+confu_matrix(3,2));
disp(FPS_CA(i,:));
disp(FPS_MG(i,:));
disp(FPS_ZN(i,:));
end

mean_total = mean(ACCS);
std_total = std(ACCS);

mean_tp_ca = mean(TPS_CA);
std_tp_ca = std(TPS_CA);
mean_tp_mg = mean(TPS_MG);
std_tp_mg = std(TPS_MG);
mean_tp_zn = mean(TPS_ZN);
std_tp_zn = std(TPS_ZN);
mean_fp_ca = mean(FPS_CA);
std_fp_ca = std(FPS_CA);
mean_fp_mg = mean(FPS_MG);
std_fp_mg = std(FPS_MG);
mean_fp_zn = mean(FPS_ZN);
std_fp_zn = std(FPS_ZN);

fprintf('\n\n%g %g %g %g %g %g %g\n',
mean_total,mean_tp_ca,mean_tp_mg,mean_tp_zn,mean_fp_ca,mean_fp_mg,mean_fp_z
n );
fprintf('%g %g %g %g %g %g %g\n\n',
std_total,std_tp_ca,std_tp_mg,std_tp_zn,std_fp_ca,std_fp_mg,std_fp_zn );

disp(MATRIX);

```

```

TP_CA = MATRIX(1,1)/(MATRIX(1,1)+MATRIX(2,1)+MATRIX(3,1));
TP_MG = MATRIX(2,2)/(MATRIX(1,2)+MATRIX(2,2)+MATRIX(3,2));
TP_ZN = MATRIX(3,3)/(MATRIX(1,3)+MATRIX(2,3)+MATRIX(3,3));
disp(TP_CA);
disp(TP_MG);
disp(TP_ZN);

FP_CA =
(MATRIX(1,2)+MATRIX(1,3))/(MATRIX(1,2)+MATRIX(1,3)+MATRIX(2,2)+MATRIX(2,3)+
MATRIX(3,2)+MATRIX(3,3));
FP_MG =
(MATRIX(2,1)+MATRIX(2,3))/(MATRIX(1,1)+MATRIX(1,3)+MATRIX(2,1)+MATRIX(2,3)+
MATRIX(3,1)+MATRIX(3,3));
FP_ZN =
(MATRIX(3,1)+MATRIX(3,2))/(MATRIX(1,1)+MATRIX(1,2)+MATRIX(2,1)+MATRIX(2,2)+
MATRIX(3,1)+MATRIX(3,2));
disp(FP_CA);
disp(FP_MG);
disp(FP_ZN);

end

```

Anexo 4

1) Escalas de hidrofobicidad

Tipo de aminoácido	Hidrofobicidad 1	Hidrofobicidad 2	Hidrofobicidad 3
Ile	4,5	0,31	-0,6
Val	4,2	-0,07	-0,31
Leu	3,8	0,56	-0,55
Phe	2,8	1,13	-0,32
Cys	2,5	0,24	-0,13
Met	1,9	0,23	-0,1
Ala	1,8	-0,17	0,11
Gly	-0,4	-0,01	0,74
Thr	-0,7	-0,14	0,52
Ser	-0,8	-0,13	0,84
Trp	-0,9	1,85	0,3
Tyr	-1,3	0,94	0,68
Pro	-1,6	-0,45	2,23
His	-3,2	-0,96	2,06
Glu	-3,5	-2,02	2,68
Gln	-3,5	-0,58	2,36
Asp	-3,5	-1,23	3,49
Asn	-3,5	-0,42	2,05
Lys	-3,9	-0,99	2,71
Arg	-4,5	-0,81	2,58

Anexo 5

- 1) Parámetros óptimos para los modelos basados en SVM y diversos tipos de información.

Información	C óptimo	γ óptimo
Evolutiva	128	0,1250
Geométrica	2048	0,0078
Fisicoquímica	64	0,0313
Evo+Geo	256	0,0078
Evo+Fis	1024	0,0039
Geo+Fis	2048	0,0039
Evo+Geo+Fis	2048	0,0010

- 2) Parámetros óptimos para los modelos de SVM con reducción de dimensionalidad.

Información	C óptimo	γ óptimo
Evo+Geo+Fis SC	2048	0,0020
Evo+Geo+Fis SCSR	2048	0,0020

Anexo 6

Atributos Evolutivos (3):

Puntajes de PSSM = 3

Atributos Geométricos (36):

Distancias = 20

Ángulos = 16

Atributos Fisicoquímicos (62):

Hidrofobicidad = 12

Composición atómica = 24

Conteo de átomos = 6

Energías = 15

Área accesible al solvente = 1

Composición aminoacídica = 4