Institución Universitaria

Metodología para la selección de la métrica de distancia en Neighborhood Kernels para clasificación semi-supervisada de secuencias proteicas

Norma Patricia Guarnizo Cutiva

Instituto Tecnológico Metropolitano

Facultad de Ingenierías

Medellín, Colombia

2016

Metodología para la selección de la métrica de distancia en Neighborhood Kernels para clasificación semi-supervisada de secuencias proteicas

Norma Patricia Guarnizo Cutiva

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Magister en Automatización y Control Industrial

Director (a):

PhD., MSc. Jorge Alberto Jaramillo Garzón.

Línea de Investigación:

MIRP - Máquinas Inteligentes y Reconocimiento de Patrones

Grupo de Investigación:

AECC – Automática Electrónica y Ciencias Computacionales

Instituto Tecnológico Metropolitano

Facultad de Ingenierías

Medellín, Colombia

2016

Dedicatoria

A mis Hijas, Isabel y Valeria, y a mis Padres, Olga y Javier.

Agradecimientos

Desde el punto de vista personal, agradezco a mis hijas, Isabel y Valeria, por haber comprendido el sacrificio que un trabajo como este conlleva, por apoyarme y soportar con paciencia y amor mis ausencias, a Fabio Saldarriaga por cuidar con amor de ellas, y a mis padres, porque siempre han sido mi soporte, con quienes puedo contar frente a cualquier adversidad.

A mi Director de Tesis, PhD. Jorge Alberto Jaramillo Garzón, por haber aceptado ser mi director, por tenerme paciencia en estos largos años y no declinar, por compartir conmigo su vasto conocimiento en bioinformática y en reconocimiento de patrones, por sus enseñanzas, pero sobre todo, por ser mi amigo y apoyarme en los momentos difíciles. A los Ingenieros Mecatrónicos Juan Camilo Pineda y Andrés Felipe Cardona, por formar parte de este trabajo con sus valiosos aportes en el desarrollo y ejecución de algunos algoritmos bajo la tutoría del Director Jorge Jaramillo Garzón. Son muchas personas a quienes quiero agradecer, entre ellos, Julio Alberto Casas, por su confianza y apoyo durante este tiempo, a Karen Durango, por su valiosa colaboración, a mis compañeros del Departamento de Mecatrónica y Electromecánica, a mis amigos del Departamento de Electrónica y Telecomunicaciones, a mis compañeros del laboratorio MIRP, y en general, a todos aquellos que siempre estuvieron pendientes de la evolución de este trabajo y me manifestaron sus buenos deseos para que lo llevara a feliz término.

Resumen

Este trabajo presenta una metodología para la selección de métricas de distancia, entre Geométricas y Bio-inspiradas, en un clasificador semi-supervisado de máquinas de vectores de soporte (SVM), para la clasificación de secuencias proteicas de plantas terrestres (base de datos Embryophyta). Primero se construyó una matriz kernel mediante un proceso de extracción y selección de características, por otro lado, se construyó una matriz para las distancias Euclídea, Mahalanobis, Mismatch y Gappy. Ambas matrices fueron usadas en el algoritmo Neighbordooh kernel para obtener una matriz semi-supervisada para un clasificador SVM optimizado con PSO y W-SVM, cuyo modelo de predicción fue evaluado calculando la matriz de confusión entre los datos de entrenamiento y los datos de prueba obtenidos mediante validación cruzada, posteriormente se calcula la media geométrica con base en la sensibilidad y la especificidad. Los resultados demuestran que la metodología presentada es eficiente para seleccionar la métrica de distancia apropiada según la función molecular. La métrica Euclídea fue seleccionada como la de mejor desempeño para siete funciones, con porcentajes de acierto que van desde 49.94% hasta el 74.3%. Mismatch por su parte, fue seleccionada para tres funciones, con desempeños desde 51.63% hasta 80.78%, y por último, Gappy fue seleccionada para cuatro funciones, con aciertos desde 43.11% hasta 68.5%. Para terminar, es importante resaltar que este proyecto de investigación permitió la creación de la línea de investigación en algoritmos bioinformáticos en el ITM, además derivó cuatro trabajos de grado de pregrado y dos nuevos estudiantes de la Maestría en Automatización y Control Industrial.

Palabras clave:

Clasificador, Euclídea, Mahalanobis, PSO, Selección de características, SVM, W-SVM.

Abstract

This Project presents a methodology to select between Geometric and Bio-inspired distance metrics in a semi-supervised classifier using Support Vector Machine (SVM) to classify protein sequences from land plants (Embryophyta dataset). First, a kernel matrix was built in a process of extraction and feature selection, on the other hand, another matrix was built to Euclidean, Mahalanobis, Mismatch and Gappy distances. Both matrices were used in the Neighborhood kernel algorithm to obtain a semi-supervised matrix to an optimized SVM classifier using PSO and W-SVM. The prediction model was evaluated calculating a confusion matrix between training data and test data, with partitions from cross-validation method; after was calculated a geometric mean with the sensitivity and specificity. The results show that the methodology presented is efficient to select the best metric according to the molecular function. The Euclidean metric was selected as the best one for seven functions, with score from 49.94% to 74.3%. Mismatch was selected for three functions, with score from 51.63% to 80.78%, and Gappy was selected for four functions, with score from 43.11% to 68.5%. On the other hand, it is important to stand out that this work allowed to create a new research line in Bioinformatic algorithm in the ITM, in addition, this one derived four Degree works in Engineering and two new students of Maestría en Automatización y Control industrial.

Keywords:

Classifier, Euclidean, feature selection, Mahalanobis, PSO, SVM, W-SVM.

Contenido

Lista de figuras

Pág.

Lista de tablas

Pág.

Introducción

El aprendizaje de máquina es un área de investigación que ha tomado gran fuerza en los últimos años, debido a que a través de éste se puede conseguir la automatización en la toma de decisiones en diferentes áreas de la ciencia a partir de un análisis inteligente de datos. Esta rama de la Inteligencia artificial ha hecho importantes aportes en diferentes disciplinas como son: desarrollo de herramientas de apoyo diagnóstico para la identificación de patologías (Longstaff, Reddy, & Estrin, 2010), La predicción y alerta temprana de enfermedades de los cultivos vegetales y plagas de insectos (Li, Yang, Peng, Chen & Luo, 2008), reconocimiento de escritura (Ball & Srihari, 2009), la detección y clasificación de material web (Cheng & Li, 2006), y en la predicción de proteínas con estructuras desordenadas (Shimizu, Muraoka, Hirose, Tomi & Noguchi, 2007), entre otros.

Las dimensiones de las bases de datos de secuencias proteicas actuales son tales que la anotación manual (o etiquetado de secuencias) se ha convertido en un proceso casi intratable, generando una brecha cada vez más grande entre la cantidad de secuencias proteicas disponibles y la cantidad de proteínas anotadas (Pandey, 2006). Las secuencias biológicas de plantas terrestres, contienen poca información genómica y cuentan con un número limitado de muestras con función conocida por lo que es difícil encontrar una relación estrecha entre los organismos modelo y el organismo en estudio, filogenéticamente hablando (Giraldo-Forero et al, 2013). Como consecuencia, se obtiene un número considerable de falsos positivos y negativos, que a su vez, dificulta ampliamente el proceso de anotación de las secuencias y su posterior uso en ensayos biológicos, generando retrasos y sobrecostos en los procesos. Estos inconvenientes pueden ser superados a través del uso de técnicas de aprendizaje de máquina que impliquen el aprovechamiento, tanto de los datos etiquetados como de los no etiquetados, que pueden encontrarse con mayor facilidad en bases de datos públicas, mayoritariamente distribuidas a través de internet (Zhou & Li, 2009). Este tipo de técnicas se denominan “semi-supervisadas” y constituyen la base para el desarrollo del presente trabajo.

Algunos trabajos han utilizado kernels semi-supervisados. En (Mei & Fei, 2010) se clasifican aminoácidos usando spectrum kernel para la localización de proteínas sub-nucleares, en (Hannagan & Grainger, 2012) se propone, desde el análisis de proteinas, el uso de string kernels para establecer la forma en que el cerebro codifica la información ortográfica durante la lectura. Los kernels semi-suervisados fueron diseñados para la predicción de funciones proteicas como lo proponen (Weston et al., 2004, 2005; Leslie et al.). Estos autores proponen dos métodos conocidos como Neighborhood Kernel y Bagged Kernel, los cuales tienen como idea principal, cambiar la métrica de distancia del tal manera que la distancia relativa entre dos puntos es mucho más pequeña si los puntos están en el mismo clúster, la idea clave es el cambio del espacio de representación del clasificador, de tal manera que se haga un mejor uso de los datos no etiquetados y de su alta dimensión, comparados con los datos de los cuales se tiene certeza.

El aporte importante que hace este trabajo en la predicción funcional de proteínas de organismos vegetales, es desarrollar una metodología que permita seleccionar la métrica de distancia más apropiada, con lo cual se mejora el porcentaje de acierto en la clasificación con conjuntos de datos desbalanceados, con una alta dimensión de datos sin etiquetar y que presentan un problema de clasificación multi-clase. La métrica de distancia permite establecer la relación que existe entre dos proteínas, de tal manera que su cercanía define su similitud, con lo que se pueden agrupar por funciones biológicas.

Para el desarrollo de la metodología, primero se realizará la construcción de un espacio de representación que permita el tratamiento estadístico de los datos obtenidos mediante la extracción de características físico-químicas hecho a secuencias proteicas de organismos vegetales. En la siguiente etapa se aplicarán diferentes métricas de distancia geométricas, como la Euclídea y la Mahalanobis, así como métricas de similitud de secuencias usando kernels de secuencias, como Mismatch y Gappy, en la construcción del clasificador. Posteriormente, se evaluará el modelo de predicción del clasificador mediante el método de validación cruzada, en donde se forman particiones de los datos para obtener un conjunto para el entrenamiento y otro para la validación, con esto se logra independencia entre los conjuntos de datos y se obtiene los parámetros óptimos para el modelo. Por último, se evalúa el porcentaje de acierto del modelo con cada métrica,

mediante el cálculo de la matriz de confusión y la media geométrica. Como producto de la investigación se entregará una metodología de selección de la métrica de distancia para la construcción del clasificador semi-supervisado SVM basado en Neighborhood Kernels.

1. Preliminares

1.1 Justificación

Una de las áreas de aplicación en donde el aprendizaje semi-supervisado toma gran importancia, es la Bioinformática, una disciplina que se enfoca en el uso y desarrollo de técnicas computacionales para el tratamiento de problemas en la biología molecular y otras disciplinas asociadas. La cantidad de datos de secuencias obtenidos con el estudio de la genómica y la proteómica, ha crecido de manera exponencial, especialmente desde la finalización del proyecto de secuenciación del genoma humano. Las dimensiones de las bases de datos actuales son tales que la anotación manual (o etiquetado de secuencias proteicas) se ha convertido en un proceso casi intratable, generando una brecha cada vez más grande entre la cantidad de secuencias proteicas disponibles y la cantidad de proteínas anotadas (Pandey, 2006), lo cual ha impulsado el desarrollo de la Bioinformática a nivel mundial evidenciado por la creciente aparición de publicaciones científicas especializadas en el área, como son: “Bioinformatics” (Oxford University Press) o “BMC Bioinformatics” (Biomed Central), así como una creciente cantidad de artículos sobre bioinformática publicados en revistas con el más alto impacto científico como “Nature” o “Science”.

A nivel nacional la Bioinformática ha sido reconocida como disciplina madura e independiente, donde COLCIENCIAS está haciendo grandes esfuerzos para financiar proyectos de investigación por ser considerada un área estratégica a nivel nacional, e incluso impulsó y financió, junto a compañías como Microsoft, la puesta en marcha del Centro Nacional de Bioinformática y Biología Computacional, en el que además trabajan instituciones como la Universidad Nacional de Colombia sede Manizales y la Universidad de Caldas, entre otras. Otro de los avances del país en esta disciplina, es la creación del Centro Nacional de Secuenciación, que se vislumbra como una fuente de datos básica para el desarrollo de la bioinformática, donde no sólo se llevará a cabo el procesamiento y análisis de datos, sino que además se podrán diseñar nuevas y más eficientes herramientas de cómputo.

Colombia cuenta con una de las biodiversidades más altas del mundo, ubicándose entre uno de los doce países en los que se encuentra el 70% de las especies (Cerón, et. al., 2008). Con base en esta riqueza existe una Política Nacional de Biodiversidad de la República de Colombia (Ministerio del Medio Ambiente, 1995) que se fundamenta en que la biodiversidad es vital para la existencia del ser humano, es patrimonio de la nación y tiene un valor estratégico para el desarrollo. Por otro lado, en los documentos sobre Política Nacional de Ciencia, Tecnología e Innovación (CONPES 3582 y CONPES, 2009) se enfatiza en la Biodiversidad como uno de los cinco ejes principales para el desarrollo del país en el marco de la Política Nacional para el Fomento a la Investigación y la Innovación. Con el fin de consolidar el conocimiento de la biodiversidad del país, se deben adoptar acciones de carácter investigativo que suponen retos interesantes en las áreas de la genómica, transcriptómica y proteómica; sin embargo para que esta investigación sea rentable, es de vital importancia contar con herramientas computacionales que puedan sistematizar este conocimiento e impulsar la implementación de nuevas aplicaciones (Barreto, 2008).

Por todo lo anterior, no solamente se debe propiciar y financiar la investigación en el área de la bioinformática desde el aprendizaje de máquina en el Instituto Tecnológico Metropolitano, que ha mostrado ser competitiva a nivel nacional independientemente de la experimentación biológica que se realiza en el país, sino que, para hacer más competitiva la generación de conocimiento, debe ser integrada activamente a las líneas de investigación y así fortalecer la investigación y la economía nacional. Uno de los principales ejes que se ha planteado para el desarrollo del país en los próximos años, es la explotación de su biodiversidad. Con este fin, cada vez toma más fuerza el uso de herramientas bioinformáticas para el análisis de secuencias genómicas y proteómicas, evidenciándose en las iniciativas gubernamentales para financiar proyectos de este tipo. No obstante, los métodos actuales comúnmente utilizados para la anotación de secuencias biológicas no son aptos para la predicción en organismos vegetales, y teniendo en cuenta que estos organismos son parte esencial de la biodiversidad del país, es necesario contar con nuevas y mejores herramientas de análisis a nivel proteómico, gracias a que las herramientas de aprendizaje de máquina diseñadas con el propósito de realizar predicciones sobre organismos vegetales, presentan muy bajo rendimiento en comparación con otras herramientas diseñadas sobre organismos modelo; esto es en parte, debido a la escasa cantidad de muestras etiquetadas que resultan insuficientes para

entrenar algoritmos supervisados de manera que tengan una alta capacidad de generalización.

1.2 Hipótesis

El uso de técnicas basadas en la extracción de atributos físico-químicos y estadísticos de las secuencias de aminoácidos que representan la estructura primaria de las proteínas, permitirá representarlas en espacios de características en los cuales se puedan definir diferentes tipos de métricas de distancia para mejorar el desempeño de los clasificadores basados en Neighborhood kernels cuando son aplicados sobre organismos con poca información filogenética.

1.3 Objetivos

1.3.1 Objetivo general

Proponer una metodología de selección de la métrica de distancia en la construcción de clasificadores semi-supervisados basados en Neighborhood Kernels sobre espacios de representación basados en extracción de características para la clasificación de secuencias proteicas de plantas terrestres (embryophyta), pertenecientes a la base de datos pública Uniprot.

1.3.2 Objetivos específicos

- Construir un espacio de representación mediante extracción de características de secuencias proteicas que contenga las variables más discriminantes para la clasificación.

- Aplicar diferentes métricas de distancia definidas sobre bases de datos artificiales, en la implementación de un clasificador basado en Neighborhood Kernels para secuencias proteicas.

- Validar las métricas de distancia sobre el espacio de características construido mediante la implementación de un sistema de clasificación de secuencias proteicas obtenidas de bases de datos públicamente disponibles, que permitan ajustar los criterios de selección de dichas métricas.

2. Marco Teórico

2.1 Aspectos Biológicos

2.1.1 Propiedades físico-químicas de los Aminoácidos

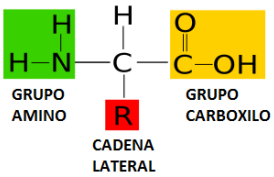
Los aminoácidos son de vital importancia en el metabolismo de los seres vivos, aparte de construir bloques de proteínas, son responsables del crecimiento, de la reparación de tejidos y de la energía necesaria de los organismos. Son ellos los encargados de hacer, entre otras funciones y dependiendo del organismo, que algunos nutrientes como el agua, grasas, minerales, carbohidratos y vitaminas liposolubles, se almacenen óptimamente en ellos (Pardo, 2004).

Las propiedades químicas de los aminoácidos determinan la actividad biológica de la proteína, y aunque existes centenares de ellos, son solo 20 aminoácidos los que pueden ser sintetizados por los organismos vivos, combinándose para formar las proteínas, dándole una gran versatilidad química y formando secuencias determinadas por el gen que codifica dicha proteína, como se muestra en la Tabla 1. Las proteínas contienen, dentro de su secuencia de aminoácidos, la información necesaria para determinar la forma en que ésta se pliega en una estructura tridimensional dando paso a cuatro estructuras y determinando la función biológica de la misma.

La composición química de los aminoácidos se representa por un grupo carboxilo o ácido (-COOH – amarillo en la figura) y un grupo amino ($-NH_2$ – verde en la figura) que se unen a un carbono α(-C-), ocupando así dos valencias del mismo, las otras dos son ocupadas por un átomo de hidrogeno (-H) y un grupo químico variable llamado radical (R – rojo en la figura) (Olivares-Quiroz & García-Colín, 2004).

Figura 1: Estructura de los Aminoácido.

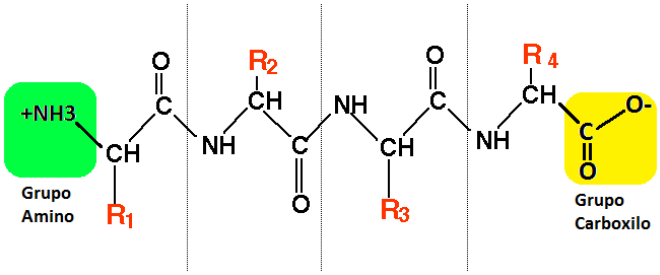
En esta figura se muestra la estructura molecular general de los aminoácidos, que se compone de un grupo amino NH_2 (de carácter básico), un grupo carboxílico COOH (de carácter ácido) y una cadena lateral "R" que es en la que difieren todos los aminoácidos.



Dos aminoácidos pueden unirse a través de la formación de un enlace peptídico, el cual consiste en la unión del grupo carboxilo de un aminoácido con el grupo amino del aminoácido adyacente, a su vez una cadena de enlaces peptídicos constituyen una cadena polipéptida como se observa en la Figura 2.

Figura 2. Cadena Polipéptida – Estructura primaria.

En esta figura se observa el código genético de la estructura primaria de una proteína. Se compone por una cadena lineal de un grupo de aminoácidos unidos por enlaces peptídicos que comienza con el grupo amino, seguido por un conjunto de aminoácidos, y termina en el grupo carboxilo.



Los grupos carboxilo y amino de los aminoácidos se ionizan en solución a pH fisiológico neutral, con el grupo carboxilo que lleva una carga negativa (-COO 2) y el grupo amino de una carga positiva (-NH3). El estado de ionización varía con el pH: en soluciones ácidas el grupo carboxilo no está ionizado (-COOH) y el grupo amino está ionizado (-NH3). Por el contrario, en soluciones alcalinas el grupo carboxilo está cargado negativamente (-COO) y el grupo amino no está ionizado (-NH2).

En general, las propiedades de ionización de los aminoácidos constituyentes, incluyendo la de sus cadenas laterales, influyen en gran medida la solubilidad, la estabilidad y la organización estructural de la proteína. Similarmente, la hidrofobicidad /hidrofilicidad de las cadenas laterales determinan un papel importante en el comportamiento fisicoquímico de la cadena polipéptida y su plegado en las estructuras tridimensionales de la proteína.

Los aminoácidos se clasifican de acuerdo con la naturaleza química (aliafática, aromática, heterocíclica) de sus cadenas laterales "R" en subclases apropiadas, sin embargo también pueden clasificarse de acuerdo a la polaridad del residuo o grupo "R" debido a que determina la posible función que tendrá el mismo en la proteína, así como sus posibles contribuciones al plegamiento de la cadena polipeptídica (Conn et al, 2005). Los aminoácidos se clasifican según su polaridad, en:

- No polares o hidrófobos (Alanina, Metionina, Fenilalanina, Prolina, Isoleucina, Triptófano, Leucina y Valina).
- Polares pero sin carga (Asparagina, Tirosina, Cisteina, Treonina, Glicina, Serina y Glutamina).
- Polares debido a una carga negativa al pH fisiológico neutro (Ácido aspártico y Ácido glutámico).
- Polares debido a una carga positiva al pH fisiológico (Arginina, Histidina y Lisina).

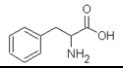
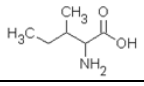
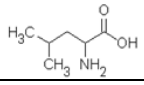
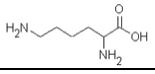
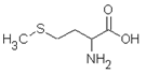
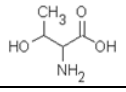
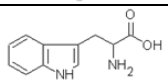
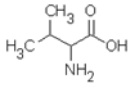
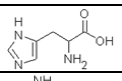
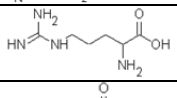
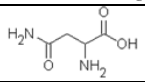
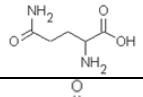
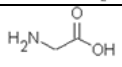
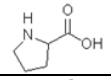
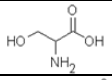
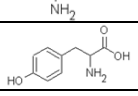
El **punto isoeléctrico (pI)** de un aminoácido corresponde al pH al cual la molécula carece de carga eléctrica y su solubilidad es prácticamente nula, lo que hace que la molécula se disocie por igual en ambos sentidos. El punto isoeléctrico se calcula como la media geométrica entre dos *pKa*, siendo estos términos la medida de la fuerza que tienen las moléculas de disociarse. Cada aminoácido tiene tres *pKa,* correspondientes a cada uno de los grupos que lo conforman (grupo carboxilo, grupo amino y grupo R).

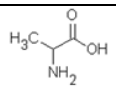
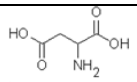
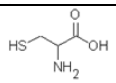
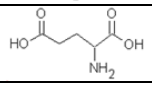
El **índice de hidropatía** en los aminoácidos corresponde a la escala que combina la hidrofobicidad y la hidrofilicidad de las cadenas laterales, se usa para predecir la tendencia de los aminoácidos a buscar un ambiente acuoso (valores negativos) o un ambiente hidrófobo (valores positivos).

La **probabilidad de ocurrencia** de un aminoácido en una proteína fue obtenida al estudiar un conjunto considerable de proteínas.

Tabla 1: Estructura de los aminoácidos.

Lista de 20 aminoácidos codificados por genes. Se presentan el nombre, su abreviatura, símbolo y estructura química. Allí se puede observar cómo difieren entre ellos por su cadena lateral "R".

Aminoácido	Abreviatura	Símbolo	Estructura química
Fenilalanina	**Phe**	**F**	
Isoleucina	**Ile**	**I**	
Leucina	**Leu**	**L**	
Lisina	**Lys**	**K**	
Metionina	**Met**	**M**	
Treonina	**Thr**	**T**	
Triptófano	**Trp**	**W**	
Valina	**Val**	**V**	
Histidina	**His**	**H**	
Arginina	**Arg**	**R**	
Asparagina	**Asn**	**N**	
Glutamina	**Gln**	**Q**	
Glicina	**Gly**	**G**	
Prolina	**Pro**	**P**	
Serina	**Ser**	**S**	
Tirosina	**Tyr**	**Y**	

Alanina	**Al**	**A**	
Ácido Aspártico	**Asp**	**D**	
Cisteína	**Cys**	**C**	
Ácido Glutámico	**Glu**	**E**	

2.1.2 Modelo de estudio "Embryophyta"

Los seres vivos se clasifican según el tipo de células que contiene la especie, siendo las eucariotas aquellos organismos que poseen núcleo, membrana (envoltura nuclear) que posee el ADN nuclear, y las procariotas, aquellos organismos cuyas células no poseen un núcleo separado. Son cinco reinos en los que se clasifican los seres vivos, según ancestros comunes o parentesco evolutivo, ellos son: moneras (bacterias), Protistas (algas), Fungi (hongos), animal y vegetal (Nabors & González-Barreda, 2004).

Las plantas tienen un conjunto de características generales que las diferencian de otros seres vivos, como son:

- Son organismos eucariotas pluricelulares.
- Utilizan energía lumínica a través de la fotosíntesis, y son autótrofas porque se alimentan a sí mismas.
- Poseen paredes celulares compuestas principalmente por celulosa, que a su vez contiene moléculas de glucosa.
- Se reproducen sexual y asexualmente en dos fases adultas.

Las plantas Embriofitas pertenecen al clado o grupo monofilético formato por los descendientes de ciertas especies de algas verdes que han sufrido algunas adaptaciones para la vida fuera del agua y que forman parte de las especies de plantas que colonizaron la tierra. Este clado comprende las plantas hepáticas, los antoceros, los musgos, los helechos y las plantas con semilla.

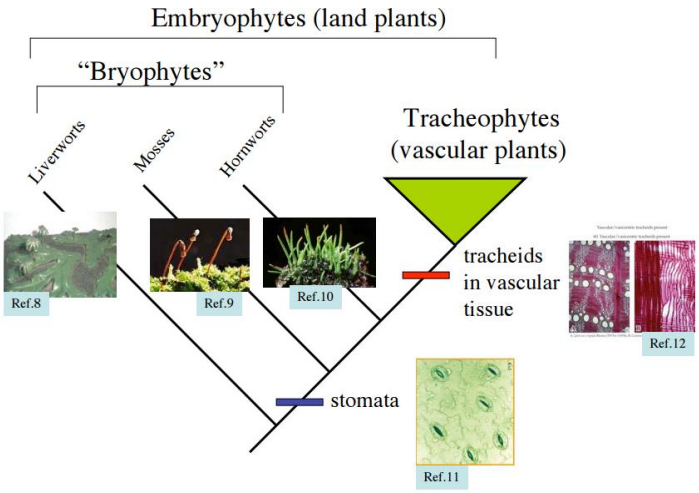
En la Figura 3 se clasifican las Embryophytes en vasculares (Tracheophytes) y no vasculares (Bryophytas). Las Bryophytes son las plantas terrestres más primitivas, son plantas pequeñas sin flores, dependen del agua para la fecundación, producen clorofila a y b, tienen paredes celulares con celulosa, usualmente tienen un crecimiento muy lento, no tienen un tejido vascular bien desarrollado, viven cerca del suelo, absorben agua por capilaridad, tienen formas especiales de crecimiento para tener un buen suministro de agua y minerales del suelo y evitar perderlos (Holman, 1961). Los antoceros, musgos y hepáticas forman parte de esta clasificación (Magallóna & Hilub, 2009).

Las plantas vasculares (Tracheophytes) poseen un tejido vascular muy organizado y eficiente, consistente en células unidas en tubos que transportan agua y nutrientes a lo largo del cuerpo del vegetal. Son en general más grandes que los Bryophytas y varían entre diminutas y gigantescas. Se clasifican en plantas con semilla (Gymnosperms y Angiosperms) y plantas sin semilla. Las plantas sin semilla se clasifican en Helechos y plantas afines, son plantas que habitan en regiones húmedas porque sus espermatozoides poseen estructuras microscópicas en forma de cola que deben nadar a través de una capa de aguar para llegar a las ovocélulas (Nabors & González-Barreda, 2004).

En la Figura 4 está la clasificación de las plantas con semilla. La Gimnospermas son aquellas que no tienen flores y habitan regiones frías cercanas a los polos y en las montañas, las coníferas como los abetos, los pinos y la secuoya pertenecen a esta clase y se caracterizan porque sus semillas se desarrollan dentro de las piñas o están a la vista sobre la hoja fértil. En las Angiospermas, que son las que tienen flores, las semillas están contenidas en el ovario, que al madurar se convierte en un fruto. Esta especie es la más predominante en nuestros tiempos, de ellas existen 20 veces más tipos que de helechos y coníferas. Se caracterizan por adaptarse exitosamente a múltiples entornos porque poseen un sistema eficiente para transportar el agua a lo largo del cuerpo vegetal, también porque la formación de un tubo polínico las liberó de su dependencia del agua para la fertilización y porque la semilla es una estructura extremadamente eficiente para la reproducción y multiplicación, ya que puede retener su vitalidad por muchos años (Holman, 1961).

Figura 3. Plantas Terrestres (Embryophytes).

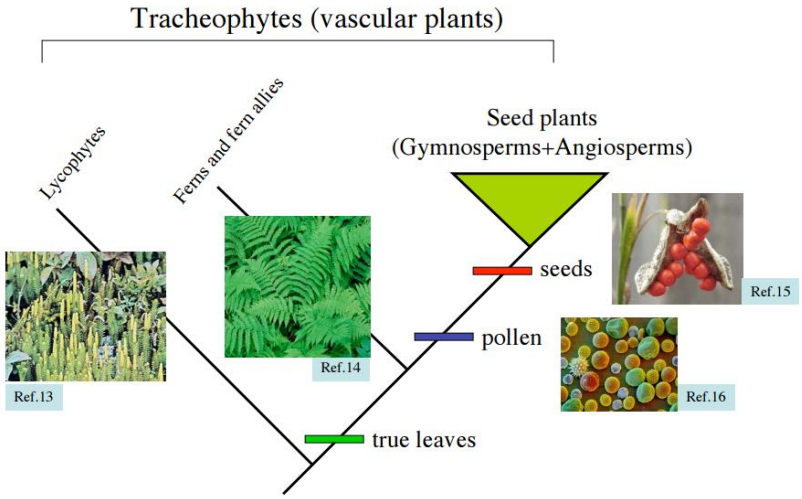
En esta figura se muestra la clasificación de las plantas terrestres Embryophyta, allí se clasifican en Vasculares (Tracheophytas) y no vasculares (Bryophytas).



Fuente:http://courses.washington.edu/bot113/summer/2008/LectureNotes/Lecture_Week_2-2.pdf

Figura 4. Plantas Terrestres Embryophytas (Vasculares).

En esta figura se muestra la clasificación de las plantas vasculares, divididas entre plantas sin semilla y con semillas (Gymnosperms y Angiosperms).



Fuente:http://courses.washington.edu/bot113/summer/2008/LectureNotes/Lecture_Week_2-2.pdf

En la Tabla 2 se describen las principales funciones de los aminoácidos en las plantas, describiendo los más esenciales en el metabolismo vegetal ().

Tabla 2. Funciones de los aminoácidos en las plantas.

Se describen las funciones que cumplen los aminoácidos en el metabolismos vegetal.

Aminoácido	**Función en la planta**
Leucina	Incrementa la producción, ayudando en la fecundación y amarre de fruto, y mejora la calidad del fruto.
Lisina	Interviene en mecanismos de resistencia a las tensiones externas y potencia, al igual que Alanina, la síntesis de clorofila.
Metionina	Precursor de etileno, incrementa calidad y producción. Aplicando al suelo favorece el crecimiento radical.
Valina	Interviene en mecanismos de resistencia bajo condiciones adversas.
Arginina	Estimula el crecimiento de las raíces, junto con METIONINA, teniendo una acción rejuvenecedora en la planta.
Glicina	Interviene en la síntesis de las porfirinas, pilares estructurales de la clorofila y los citocromos, siendo el principal aminoácido con acción quelatante, favoreciendo la formación de nuevos brotes.
Prolina	Tiene un papel fundamental en el equilibrio hídrico de la planta. Mantiene la fotosíntesis en condiciones adversas. Se acumula considerablemente bajo tensiones ambientales, pudiéndose incrementar hasta 25 veces de los normales, bajando ARGININA y SERINA. Aumenta el por ciento de germinación del grano de polen, sobre todo bajo temperaturas adversas.
Serina	Interviene en mecanismos de resistencia bajo condiciones ambientales adversas.
Alanina	Potencia la síntesis de clorofila.
Ácido Aspártico	Interviene en casi todos los procesos metabólicos de la planta.
Ácido Glutámico	Precursor de otros aminoácidos, estimula el crecimiento y estimula los procesos fisiológicos en hojas jóvenes. Interviene en los mecanismos de resistencia a factores adversos. Vía foliar ayuda a la planta sintetizar los aminoácidos que en ese momento requiere.

2.1.3 Proteínas

Son moléculas compuestas de una o más cadenas de aminoácidos, cuya estructura define la función que cumple en un organismo (Audesirk, Audesirk & Byers, 2003). Las proteínas

catalizan y controlan casi todos los procesos celulares, determinan la forma y estructura de la célula, catalizan las reacciones orgánicas, regulan la concentración de los metabolitos, originan y regulan movimientos, interactúan con otros componentes biológicos desde los iones hasta moléculas complejas como ácidos nucleicos, carbohidratos, grasas y otras proteínas (Conn et al, 2005).

Las proteínas no son otra cosa que polipéptidos largos, las cadenas que tienen pocos aminoácidos o residuos, son simplemente cadenas polipéptidas, porque son demasiado pequeñas para formar un dominio funcional.

Existen varias formas de clasificar las proteínas, entre las que están:

- Por localización en un organismo: intracelular / extracelular.
- Por función: estructural/actividad biológica.
- Por forma: globular/fibrosa.
- Por composición química: simples/complejas.

En éste trabajo se estudiarán las proteínas en las plantas terrestres (Específicamente la base de datos Embryophyta) y el tipo de función biológica que cumplen desde el punto de vista de sus estructuras y propiedades físico-químicas generales, que son determinadas por la secuencia de aminoácidos que componen su secuencia primaria. En trabajos como (Avalos & Pérez-Urria, 2011) se estudian los aminoácidos que componen las proteinas de las plantas y su aporte para el crecimiento y desarrollo normal de las mismas,

2.1.4 Estructuras de las proteínas

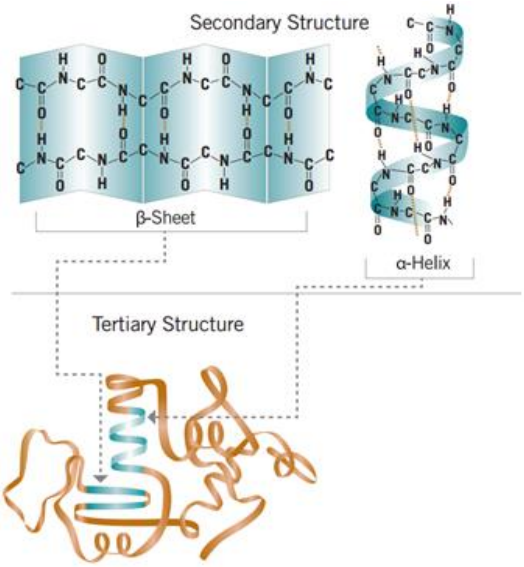
La estructura de las proteínas dependen de las fuerzas débiles que se encargan del mantenimiento de una proteína en el estado plegado: puentes de hidrógeno, interacciones hidrofóbicas, interacciones electrostáticas y las fuerzas intermoleculares de Van der Waals. La estructura primaria se refiere a la secuencia lineal en la que los aminoácidos que la componen se unen de forma covalente, a través de enlaces amida, llamados

también enlaces peptídicos (Damodaran, 2000). Por lo general la estructura primaria determina las características químicas y biológicas, además especifica los niveles superiores de la estructura de la proteína (Nabors & González-Barreda, 2004).

Una cadena de polipéptido, en la estructura secundaria, podría adoptar una gran variedad de formas espirales al azar debido a la libre rotación de los átomos que componen los diferentes enlaces a lo largo de la cadena, sin embargo, debido a las condiciones biológicas, cada proteína adopta sólo una forma debido a los enlaces de hidrógeno entre cadenas de aminos, los ángulos que se forman entre los enlaces, que a su vez son limitados por la influencia del grupo funcional de una molécula en el desarrollo de una reacción química. Las estructuras secundarias más importantes que se forman, se muestran en la Figura 5: Hélice-α y Hoja-β, las otras son variaciones de Hélice-α. Las estructuras secundarias se incorporan a la estructura terciaria, proceso que se estabiliza principalmente mediante interacciones carga-carga y enlaces covalentes fuertes denominados puentes disulfuros, que se forman entre aminoácidos que contienen azufre, como la cisteína (Nabors & González-Barreda, 2004).

Figura 5. Estructuras secundaria (Hélice-α y Hoja-β) y terciaria.

En la estructura secundaria se observa como la cadena polipeptídica se pliega en el espacio adquiriendo las formas Hélice-α y Hoja-β. La estructura terciaria se conforma por varios tramos de estructuras secundarias.



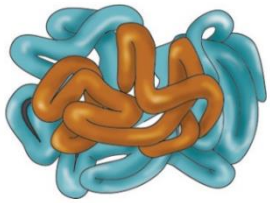
Fuente: http://www.particlesciences.com/news/technical-briefs/2009/protein-structure.html

La estructura terciaria de las proteínas comprende la disposición espacial lograda cuando una cadena polipeptídica lineal, compuesta por elementos estructurales secundarios, se pliega sobre sí misma para adquirir una forma tridimensional compacta (Damodaran, 2000). Las características comunes de la estructura terciaria revelan mucho acerca de las funciones biológicas de las mismas y sus orígenes evolutivos.

Muchas proteínas se componen de múltiples cadenas de polipéptidos, denominados subunidades de proteínas, las cuales pueden ser iguales (como en un homodímero) o diferentes (como en un heterodímero). La estructura cuaternaria se refiere a cómo estas subunidades de proteínas interactúan entre sí y se disponen para formar un complejo proteico agregado más grande. La forma final del complejo de proteína se estabiliza una vez más por diversas interacciones, incluyendo enlaces de hidrógeno, disulfuro de puentes y puentes salinos (Fuente: http://www.particlesciences.com/news/technical-briefs/2009/protein-structure.html). La estructura cuaternaria de una proteína se muestra en la Figura 6.

Figura 6. Estructura cuaternaria de la proteína.

Esta estructura se conforma por más de una cadena polipeptídica.



Fuente: http://www.particlesciences.com/news/technical-briefs/2009/protein-structure.html

2.1.5 Funciones biológicas de las proteínas

Las proteínas tienen diferentes y variadas funciones biológicas, además debido a su tamaño, forma y orientación, pueden ser clasificadas de acuerdo a su rol biológico dentro de la célula. Dependiendo de si las proteínas son homólogas, es decir, aquellas que se

han ido distanciando de un gen ancestral común, pero que tienen secuencias similares, se determina su estructura y pueden cumplir funciones biológicas similares.

Las funciones biológicas más conocidas de las proteínas son:

Proteínas Enzimáticas: las enzimas son las encargadas de catalizar diferentes tipos de reacciones químicas, éstas proteínas son las más variadas y especializadas, y han sido descubiertas en diferentes organismos.

Proteínas nutrientes y de almacenamiento: las plantas almacenan nutrientes en sus semillas, estas proteínas son importantes para el crecimiento y la germinación de la planta. Se pueden encontrar en plantas como el maíz, el trigo y el arroz, también se pueden encontrar proteínas nutrientes en el huevo, la leche y tejidos vegetales y animales.

Proteínas contráctiles o móviles: estas se encuentran en las células musculares, ayudan al movimiento, a contraer y extender favoreciendo la motricidad.

Proteínas de transporte: estas proteínas le permiten a las sustancias ser transportadas a su destino. Ejemplos de éstas son las que transportan moléculas de un órgano a otro, como el oxígeno a la sangre a través de los pulmones y haciéndolo llegar también a los tejidos del cuerpo.

Proteínas estructurales: son proteínas fibrosas que le dan soporte y forma a los organismos, una de ellas es el colágeno que se encuentran en la constitución de los tendones y cartílagos, la elastina se encuentra en los ligamentos, la queratina se encuentra en el cabello, uñas, plumas, cuernos, entre otros.

Proteínas regulatorias: son las encargadas de la regulación de las actividades de otras proteínas implicadas en la transcripción y translación. Algunas de estas son hormonas como en el caso de la insulina que regula el metabolismo del azúcar en la sangre.

Proteínas de defensa: son las que defienden los organismos contra la invasión de otras especies, ejemplo de ellas son los anticuerpos, que son especializados en reconocer, precipitar o neutralizar la invasión de microorganismos o proteínas extrañas de otras

especies, otro ejemplo es la proteína que detienen la pérdida de sangre cuando sucede un daño en el sistema vascular, otras como algunos venenos de serpiente, toxinas bacterianas y de plantas tienen funciones defensivas.

El proyecto GO se encarga de construir un vocabulario controlado y estructurado, conocido como ontologías, que se usa para ser aplicado en la anotación de secuencias proteicas, genes, o productos de genes, entre otros. The Arabidopsis Information Resource (TAIR) es un consorcio que posee una de las bases de datos biológicos más grande, especializada y confiable en el estudio de las plantas hoy en día. Alrededor de 11.000 investigadores y 4.000 organizaciones en el mundo han aunado esfuerzos y experticia para generar una rica diversidad y cantidad de información y material especializado en notación de material biológico (Rhee et al, 2003). TAIR seleccionó especialmente para plantas, un conjunto de notaciones biológicas que se agrupan, según el Gene Ontology, en tres ontologías generales, que describen atributos de elementos genéticos de cada uno de estos dominios de biología molecular. Estos tres subconjuntos son:

FUNCION MOLECULAR (Molecular function): las anotaciones de este subconjunto se refieren a la actividad principal o función del producto genético de un gen, independientemente de donde y cuando realiza dicha función. Algunos ejemplos de los niveles superiores de la jerarquía serían “enzima” o “proteína estructural”, mientras que “adenilato ciclasa” o “actina” serían ejemplos de niveles inferiores y, por tanto, funciones más específicas.

PROCESO BIOLÓGICO (Biological proccess): las anotaciones de esta ontología hacen referencia a la dinámica en que está englobada una proteína, procesos que son llevados a cabo por conjuntos ordenados de funciones moleculares. Así pueden definir rutas metabólicas o procesos celulares. Algunos ejemplos serían “mitosis” o “metabolismo de purinas”.

COMPONENTE CELULAR (Cellular component): Describe estructuras sub-celulares, localizaciones, complejos macromoleculares, en conclusión, lugares donde un proceso biológico ejerce su función. Así podríamos encontrar la anotación “citoplasma”, dentro de ella “mitocondria” y con mayor detalle “membrana mitocondrial interna”.

Un término GO describe una función molecular, un proceso biológico o un componente celular, pero no representa ni un gen ni un producto genético. Cada término del vocabulario de Gene Ontology consiste en un identificador alfanumérico único, un nombre común y una definición.

Así, un término de GO está definido por tres elementos:

1. Nombre (Ej: “hydrolase activity”)
2. Identificador único (Ej: “GO:0016787”)
3. Definición (Ej: “Catalysis of the hydrolysis of various bonds, e.g. C-O, C-N, C-C, phosphoric anhydride bonds, etc. Hydrolase is the systematic name for any enzyme of EC class 3”).

Tabla 3. GO Slim Terms.

Se relacionan los GO términos que pertenecen a la Ontología Función Molécular (GO Molecular Function) utilizada en este trabajo.

Keyword Cathegory	GO Slim Term (GO id)	Description
GO Molecular Function	hydrolase activity (GO:0016787)	Includes this term and all of its children
	kinase activity (GO:0016301)	Includes this term and all of its children
	transferase activity (GO:0016740)	Includes this term and all of its children
	other enzyme activity (GO:0003824)	Excludes hydrolase, kinase and transferase activities
	transcription factor activity (GO:0003700)	Includes this term and all of its children
	DNA or RNA binding	Includes DNA binding GO:0003677or RNA binding GO:0003723 and excludes transcription factor activity GO:0003700
	other nucleic acid binding (GO:0003676)	Excludes DNA binding GO:0003677, RNA binding GO:0003723 and transcription factor activity GO:0003700

	nucleotide binding (GO:0000166)	Includes this term and all of its children
	protein binding (GO:0005515)	
	receptor binding and activity	Includes receptor binding GO:0005102 or receptor activity GO:0004872 and all of their children
	other binding (GO:0005488)	Excludes nucleic acid binding (GO:0003676),nucleotide binding (GO:0000166),DNA binding GO:0003677, RNA binding GO:0003723,transcription factor activity GO:0003700, protein binding (GO:0005515), receptor binding GO:0005102, receptor activity GO:0004872
	structural molecule activity (GO:0005198)	includes this term and all of its children terms
	transporter activity (GO:0005215)	Includes this term and all of its children
	molecular function unknown (GO:0005554)	Genes for which the function is not known or cannot be inferred
	other molecular functions (GO:0003674)	Excludes all of the other Molecular function GO slim categories

2.2 Clasificación de secuencias proteicas

La predicción de funciones de proteínas a nivel molecular, celular y fenotípico, a partir de sus correspondientes secuencias de aminoácidos, constituye un tema fundamental para el conocimiento de las especies, pues permite que la enorme cantidad de datos almacenados se convierta en conocimiento biológico. Sin embargo, la determinación de las funciones de las proteínas requiere, en la mayoría de casos, de aproximaciones experimentales realizadas en el laboratorio, y estos procesos deben estar enfocados sobre proteínas o funciones específicas, o usar muestras de ADN o proteínas clonadas a partir de los genes de interés, lo cual hace de la experimentación un proceso altamente costoso y demorado. Esta perspectiva ha llevado a varios investigadores a proponer la predicción computacional

como herramienta confiable de análisis, para dilucidar sobre las funciones de algunas proteínas importantes (Baldi and Brunak, 2001).

En (Radivojac et al, 2013) se presenta una completa revisión y evaluación de trabajos realizados donde aplican técnicas computacionales para la predicción de funciones protéicas. Entre los trabajos desarrollados en este sentido están los numerosos métodos como GOblet (Groth et. al., 2004), OntoBlast (Zehetner, 2003), GOFigure (Khan, 2003) y GOtcha (Martin et. al., 2004), que están basados en la idea de refinar y mejorar los resultados iniciales entregados por herramientas clásicas de alineamiento de secuencias como BLAST (Basic Local Alignment Search Tool) y PSI- BLAST (Position Specific Iterative - BLAST), a través de mapeos y ponderaciones de las funciones específicas asociadas a las predicciones de BLAST. Sin embargo, en estos métodos no se tiene en cuenta la incapacidad de las herramientas tradicionales de alineamiento, para identificar adecuadamente proteínas homólogas para e-valores significativos (Hawkins et. al., 2009). El mismo problema se presenta para algunos métodos más recientes que han mejorado puntos específicos de esta metodología, como los trabajos de (Jones et. al., 2008) que se centraron en aumentar la velocidad del procedimiento mediante la inclusión de reglas de decisión, o el de (Conesa & Götz, 2008) que incluyó funcionalidad adicional para la visualización y minería de datos.

Con el fin de evitar la dependencia de los alineamientos tipo BLAST, métodos más recientes han usado herramientas de aprendizaje de máquina entrenadas sobre espacios de características físico-químicas y estadísticas. Estos métodos emplean técnicas como las redes neuronales en el caso de ProtFun (Jensen, 2003), clasificadores bayesianos (Jung & Thon, 2008) o máquinas de vectores de soporte en el caso de SVM-Prot (Cai, 2003), GOKey (Bi et. al., 2006) y PoGO (Jung et. al., 2010), obteniendo resultados con alto desempeño en sus propias bases de datos, principalmente compuestas de organismos modelo como bacterias y unas pocas especies de orden superior.

Sin embargo, cuando estos métodos son aplicados a otros organismos más relevantes para la economía y el estudio de la biodiversidad, como son las plantas terrestres, es necesario discutir una serie de aspectos. En primer lugar, al abordar mediante los métodos convencionales basados en similitud, la anotación de organismos con poca información genómica, se encuentra con una gran dificultad, por cuanto filogenéticamente no existe

una relación estrecha entre los organismos modelo y el organismo en estudio. Como consecuencia, se obtiene un número considerable de falsos positivos y negativos, que a su vez, dificulta ampliamente el proceso de anotación de las secuencias y su posterior uso en ensayos biológicos, generando retrasos innecesarios en los cronogramas de actividades y sobrecostos que afectan el presupuesto.

En segundo lugar, de los métodos descritos anteriormente, sólo Blast2GO (Conesa & Götz, 2008) está especializado en la predicción de funciones para organismos vegetales. De hecho, como los mismos autores lo plantean, muy pocos recursos están disponibles hoy en día para la anotación funcional a gran escala de especies no-modelo. Unos pocos métodos especializados en especies vegetales han sido propuestos recientemente, pero sólo realizan predicciones a nivel de componente celular, tales como Predotar (Small et. al., 2004), TargetP (Emanuelsson et. al., 2000) y Plant-mPloc (Chou and Shen, 2010). Más aún, Predotar y TargetP sólo pueden discriminar entre tres o cuatro ubicaciones sub-celulares. Plant-mPloc, mientras tanto, puede cubrir doce ubicaciones sub-celulares diferentes y fue rigurosamente probado sobre una base de datos con menos del 25% de identidad entre secuencias, donde herramientas basadas en alineamientos BLAST seguramente fallarían. Para ese conjunto de datos, los autores obtuvieron una tasa de acierto de apenas un 63.7%, muy inferior a los valores de acierto reportados por otros predictores de ubicaciones sub-celulares probados sobre bases de datos que no incluyen proteínas de plantas.

En tercer lugar, ninguno de los métodos existentes puede ser usado para tratar con proteínas que puedan existir simultáneamente en dos o más ubicaciones sub-celulares (Chou and Shen, 2010) o pertenecer a múltiples clases funcionales al mismo tiempo (Briesemeister et. al., 2010). Dado que naturalmente las proteínas pertenecen a diversas clases funcionales simultáneamente, se origina un problema muti-etiqueta que ningún método ha considerado hasta el momento. Además, en general, sólo un pequeño número de proteínas han sido actualmente anotadas para una determinada función. Por lo tanto, es difícil obtener suficientes datos de entrenamiento para un algoritmo de aprendizaje supervisado (sólo con muestras etiquetadas) (Zhao et. al., 2008). Adicionalmente, dado que la anotación funcional puede estar incompleta, es difícil justificar cuándo las proteínas asociadas a clases diferentes a la clase en estudio pueden considerarse como muestras

negativas. Consecuentemente deben ser consideradas como muestras no etiquetadas (Bi et. al., 2006). Es conveniente entonces que la metodología de clasificación empleada, sea capaz de extraer información de las muestras no etiquetadas, además de las muestras con etiqueta conocida.

Trabajos recientes han utilizado kernels semi-supervisados y técnicas de optimización para el problema de desbalance de clases. En (Mei & Fei, 2010) clasifican aminoácidos usando spectrum kernel para la localización de proteínas sub-nucleares, en (Hannagan & Grainger, 2012), por su parte, se propone un análisis proteico usando string kernels para establecer la forma en que el cerebro codifica la información ortográfica durante la lectura. En (García-López et al, 2013) presentan una optimización para el problema de desbalance de clases basado en costo de sensibilidad para la predicción de funciones biológicas. En (Weston et al., 2004, 2005; Leslie et al.) utilizan kernels semi-supervisados para la clasificación de secuencias proteicas, allí proponen dos métodos conocidos como Neighborhood Kernel y Bagged Kernel, los cuales tienen como idea principal, cambiar la métrica de distancia del tal manera que la distancia relativa entre dos puntos es mucho más pequeña si los puntos están en el mismo cluster, la idea clave aquí es el cambio del espacio de representación del clasificador, teniendo en cuenta la estructura descrita por los datos no etiquetados.

El Neighborhood Kernel utiliza el promedio sobre un vecino de secuencias definidas por una medida de similitud de secuencias locales mientras que el Bagged Kernel utiliza el agrupamiento o clustering empaquetado de las secuencias completas para modificar el Kernel base (Weston et al., 2004). Aunque estos dos métodos de Kernels semi-supervisado han ofrecido buenos resultados para la detección de secuencias homólogas, son altamente dependientes de la métrica de distancia que se use para definir la vecindad sobre la que se realizan los promedios. La métrica usada por (Weston et al., 2004) consiste en el cálculo de los E-valores obtenidos por algoritmos de alineamiento de secuencias conocidos ampliamente como BLAST y PSI-BLAST (Se et al.) con los cuales no se obtienen buenos resultados cuando no se cuenta con suficiente información filogenética de los organismos analizados.

2.3 Aprendizaje de máquina

El aprendizaje de máquina o aprendizaje automático computacional es una rama de la Inteligencia artificial que está orientada al desarrollo de algoritmos que permiten analizar datos y aprender de ellos interactivamente, de forma tal que la máquina sea capaz de generalizar comportamientos. Esta tecnología permite analizar datos de mayor volumen, con mayor complejidad y ofreciendo resultados más exactos en menor tiempo, haciendo posible la toma de decisiones más acertadas, incluso sin intervención humana.

Existen diferentes formas de aprendizaje automático: el supervisado, el no supervisado y el semi-supervisado. El aprendizaje supervisado es aquel en donde se intenta aprender de datos etiquetados, se asume que contienen características o atributos que especifican o definen a qué categoría o clase pertenecen, de un conjunto de categorías o clases predefinidas, de esta manera cada dato se asocia con su clase. Al conjunto de datos del cual se intenta aprender se le llama conjunto de entrenamiento. Este tipo de aprendizaje presenta el inconveniente de necesitar una cantidad considerable de datos (Chen & Wang, 2011) y el apoyo de expertos que se encarguen de etiquetarlos, razón por lo que es costoso, además en la mayoría de aplicaciones del mundo real, los datos obtenidos no están etiquetados y se encuentran fácilmente en bases de datos, colecciones o en internet (Zhou & Li, 2009).

En el método no supervisado, un modelo es ajustado a las observaciones, este se distingue del supervisado por el hecho de que no hay un conocimiento a priori de los datos. En este aprendizaje un conjunto de datos de objetos de entrada es tratado como un conjunto de variables aleatorias, y se construye un modelo de densidad para el conjunto de datos. El aprendizaje no supervisado puede ser usado en conjunto con la Inferencia bayesiana para producir probabilidades condicionales para cualquiera de las variables aleatorias dadas (Li & Wu, 2004).

Existe otro tipo de aprendizaje que se refiere a métodos donde automáticamente se ingresan datos sin etiquetar, y junto con los datos etiquetados, son capaces de mejorar el rendimiento en el aprendizaje sin alguna intervención humana (Chapelle, Schölkopf & Zien,

2006; Zhu & Goldberg, 2009). No obstante, diversos autores (Bodo, 2008; Chapelle & Zien, 2006) han planteado que los datos no etiquetados pueden ayudar a obtener una mejor aproximación de la función o algoritmo de decisión, mejorando así la generalización de un clasificador tradicionalmente supervisado. Existen varias técnicas para el aprendizaje con datos etiquetados y no etiquetados (Zhou, 2006), como son: aprendizaje semi-supervisado (SSL), aprendizaje transductivo y aprendizaje activo. La diferencia entre estas técnicas radica en los supuestos hechos sobre los datos de prueba.

En el aprendizaje transductivo, que tiene su origen en la teoría del aprendizaje estadístico (Vapnik, 1998) y cuyo objetivo es proporcionar una penetración importante en la explotación de los datos no etiquetados (Zhou & Li, 2009), el conjunto de datos de prueba se conoce de antemano y el objetivo del aprendizaje es optimizar la capacidad de generalización de estos datos de prueba mientras que los ejemplos no etiquetados son exactamente los ejemplos de prueba. En el aprendizaje semi-supervisado (SSL) el conjunto de datos de prueba no es conocido y los ejemplos no etiquetados no son ejemplos de prueba necesarios. Por último, en el aprendizaje activo, la explotación de los datos sin etiqueta, requiere de un experto humano, del cual las etiquetas marcadas como verdaderas puedan ser consultadas; el objetivo de este aprendizaje es reducir al mínimo el número de consultas.

El SSL ha sido implementado en numerosas aplicaciones como en salud usando los teléfonos móviles para el monitoreo de pacientes con enfermedades crónicas o en rehabilitación física, ayudándoles en la clasificación inteligente de las actividades a realizar (Longstaff & Estrin, 2010), en la agricultura para predecir la aparición temprana de plagas en los cultivos (Li, Yang, Peng, Chen & Luo, 2008), en el reconocimiento de escritura (Ball & Srihari, 2009), en el filtrado de spam en las cuentas de correo electrónico (Cheng & Li, 2006), en la predicción de desórdenes proteínicos mediante el uso de estructuras proteínicas desconocidas (Shimizu, Muraoka, Hirose, Tomii & Noguchi, 2007), entre otras.

2.3.1 Cluster assumption

En general, muchos métodos de SSL se basan en hipótesis geométricas acerca de la distribución de los datos. La primera hipótesis es la de agrupamiento, conocida como cluster assumption, esta favorece la frontera de decisión para la clasificación (Dai & Yeung, 2007), de tal manera que se ubique en una región de baja densidad, de lo contrario, una frontera de decisión en una región de alta densidad podría partir el grupo (cluster) en dos diferentes clases (Wu, Diao, Li, Fang & Ma, 2009). Ésta hipótesis asume que los datos forman grupos y que puntos ubicados en las cercanías de un grupo tienen una alta probabilidad de pertenecer a la misma clase, o lo que es lo mismo, a tener la misma etiqueta (Joachims, 1999; Chapelle & Zien, 2005; Chapelle et al., 2006; Sindhwani et al., 2006). Muchos algoritmos de clasificación se han desarrollado bajo ésta hipótesis, siendo máquinas de vectores de soporte semi-supervisadas (S^3VM), uno de los más usados.

2.3.2 Manifold assumption

Por otro lado está la hipótesis de variedades, conocida como manifold assumption, que asume los puntos de datos como la formación de una variedad de pocas dimensiones en un espacio de entrada. Bajo este supuesto se suele utilizar el grafo Laplaciano de una representación gráfica basada en la caracterización de la estructura de dicha variedad (Sindhwani, Niyogi, & Belkin, 2005). Con ésta hipótesis se presume que los datos se encuentran cerca de una variedad de baja dimensión, y la distancia intrínseca entre ellos es relevante para la clasificación. Bajo esta hipótesis se plantea que todos los datos son representados por los nodos que conforman una gráfica, los bordes de la misma son etiquetados con las distancias por parejas de los nodos incidentes, la falta de un borde, significa una distancia infinita (Wu et al., 2009).

2.4 Máquinas de Vectores de Soporte (SVM)

2.4.1 El Clasificador binario SVM

Desde el punto de vista teórico, los métodos Kernel se basan en algunas ideas de la teoría de aprendizaje estadístico introducido por Vapnik y Chervonenkis, estos han sido implementados en las máquinas de aprendizaje para lograr clasificadores como el SVM, con alta capacidad de generalización a partir de una clara intuición de lo que es el aprendizaje a partir de los datos (Vapnik, 1998).

Un clasificador binario se denota así:

$$f: R^N = X \quad (1)$$

$$f : X \to \{\pm 1\} \quad (2)$$

$$(x_1, y_1), \dots, (x_m, y_m) \in X \to \{\pm 1\} \quad (3)$$

Usando datos de entrenamiento, donde el dominio X es de carácter N-dimensional y y_i son las clases etiquetadas. Esto es, $f$clasificará correctamente nuevas muestras (x, y), las cuales provienen de la misma distribución de probabilidad P(x, y), de la cual se sacan tanto los datos de entrenamiento como los de validación. Según Vapnik, es importante restringir la clase de funciones que se implementarán en la máquina (Zhao), a una función cuya capacidad sea adecuada para la cantidad de datos disponible para el entrenamiento.

Para cualquier clasificador binario se requiere una alta capacidad de generalización, es decir, que tenga un alto porcentaje de acierto en la clasificación de las nuevas muestras y como sólo existen dos clases, pueden ocurrir dos situaciones, que la muestra sea idéntica o que sea diferente a una de las clases. El valor y de cada muestra debe ser tal que (x, y) tenga una similitud con las muestras de entrenamiento, por lo que es preciso hablar de una medida de similitud aplicada a todas las entradas x tomadas de X para llevarlas a una salida $\{\pm 1\}$. El acierto del clasificador puede evaluarse logrando un pequeño valor de riesgo, conocido como:

$$R|f| = 1/2 \int |f(x) - y| dP(x, y) \quad (4)$$

2.4.2 Clasificación linealmente posible

Un clasificador binario crea un plano que separa dos clases $\{+1\}$ y $\{-1\}$. El objetivo es encontrar un hiperplano de separación de tal forma que las muestras con la misma etiqueta, queden en el mismo lado del hiperplano como se muestra en la Figura 7. Las muestras x_i que están en el hiperplano de separación o frontera de decisión satisfacen la siguiente función lineal:

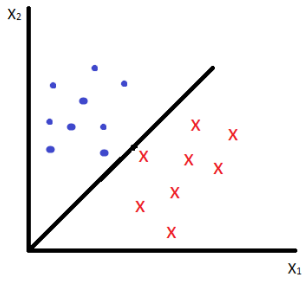
$$f(x) = (w' x_i) + b = 0 \qquad (5)$$

Los vectores $w\ y\ x$ son ortogonales entre sí:

$$(w'\ x_i\ ) = \|w\|.\|x\| Coseno(\pi/2) = 0 \qquad (6)$$

Figura 7. Clasificador binario

Hiperplano de separación entre dos clases.



El clasificador de vectores de soporte (SV) lineal se basa en la clase de hiperplanos. Si los datos son linealmente separables, entonces se tiene un vector de pesos $\boldsymbol{w}$ y un intercepto $\boldsymbol{b}$ tales que:

$$y_i.(\boldsymbol{w}' x_i) + b > 0 \quad i = 1, \dots, N \qquad (7)$$

Re-escalando $\boldsymbol{w}$ y $\boldsymbol{b}$ de tal forma que para una clase $(\boldsymbol{w}' x_i) + b = 1$ y para la otra clase $(\boldsymbol{w}' x_i) + b = -1$, como se muestra en la Figura 8. Este es el principio de un clasificador SVM, crear una frontera de decisión lo suficientemente amplia para maximizar la distancia entre los dos planos formados, sujetos a la condición del signo atribuido a cada clase, de tal forma que se cumpla:

$$y_i.(\boldsymbol{w}' x_i) + b \geq 1 \quad i = 1, \dots, N \qquad (8)$$

Esta función de decisión corresponde a una clasificación linealmente posible, es decir, no es necesario representar los datos en un espacio de dimensión superior, por lo que el discriminante es simplemente una función lineal, por otro lado, cuando es imposible la clasificación lineal, aún implementado parámetros de sintonización como se verá en el numeral 3, entonces se habla de un SVM no lineal, y su función de decisión será:

$$y_i.(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) \geq 1 \quad i = 1, \dots, N \qquad (9)$$

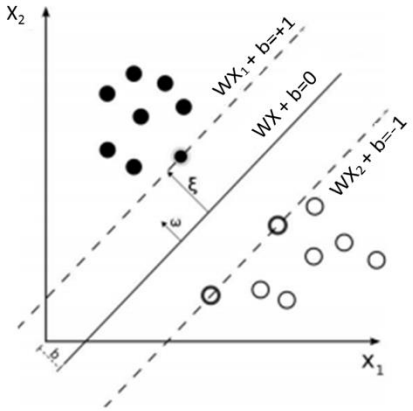
La norma del vector de pesos $\boldsymbol{w}$:

$$\|w\|^2 = w'.w \qquad (10)$$
$$\|w\| = \sqrt{w'.w}$$

Para ampliar el margen es preciso maximizar la distancia entre los planos, los cuales a su vez están formados sobre vectores de soporte, que son muestras que pertenecen a una de las dos clases y son las más cercanas al hiperplano de separación principal como se muestra en la Figura 8, además son las que contienen toda la información relevante sobre el problema de clasificación.

Figura 8. Clasificador binario SVM

Clasificador SVM donde se ilustran las clases x_1 y x_2 separadas por el hiperplano de separación que está delimitado por los vectores de soporte.



Se debe lograr una diferencia amplia entre los dos hiperplanos formados por los vectores de soporte, cada uno perteneciente a una clase.

Plano de separación 1: donde $y_i \in \{+1\}\ \ wx_1 + b = +1$
Plano de separación 2: donde $y_i \in \{-1\}\ \ wx_2 + b = -1$
Se debe maximizar $x_1 - x_2$

$$\begin{aligned} x_1 &= \ (1-b)/\|w\| \\ x_2 &= -(1+b)/\|w\| \\ x_1 - x_2 &= 2/\|w\| \end{aligned} \qquad (11)$$

La norma de w es $\|w\|$, pero existe un inconveniente con este término porque involucra una raíz cuadrada (10), por lo que se reemplaza por la expresión $1/2\|w\|^2$, donde ½ se hace por conveniencia matemática. Para construir el hiperplano óptimo de separación es necesaria la minimización de la siguiente función:

$$Minimizar\ 1/2\|w\|^2 \qquad (12)$$

Donde $w\ \in\ \mathcal{H},\ b \in \mathbb{R}$

Para lograr esto se debe resolver un problema de programación cuadrática (QP), para lo cual existen algoritmos de optimización muy eficientes, los cuales se desarrollan sobre la teoría Lagrangiana para la minimización de funciones, sin embargo esta función debe tener un tratamiento especial debido a que se sale de los supuestos Lagrangianos iniciales, que son:

Para minimizar la función $f(x), sujeto\ a\ g(x) > 0$

En nuestro caso tenemos:

Para minimizar la función $f(x), sujeto\ a\ g(x) - 1 = 0$, se debe implementar un multiplicador de Lagrange $\propto_i \geq 0$ $(\propto := (\propto_1, \ldots, \propto_m))$ que permita cumplir esta condición. Ahora se debe simplificar la tarea de aprendizaje mediante la minimización de la siguiente expresión Lagrangiana, llamada también función prima objetivo:

$$L = (w, b, \propto) = 1/2\|w\|^2 - \sum_{i=1}^{m} \propto_i \left(y_i\left((\boldsymbol{w}' x_i) + b\right) - 1\right) \quad (13)$$

En esta ecuación $w, b\ y\ \propto_i$ tienen, en algún lugar del hiperespacio, valores tales que permitan minimizar la función con respecto a $w\ y\ b$, y maximizar con respecto a $\propto_i$, por lo que es preciso derivar parcialmente con respecto a las variables primas $w\ y\ b$, y hacer que se cumpla:

$$\propto_i \left(y_i\left((\boldsymbol{w}' x_i) + b\right) - 1\right) = 0\ para\ todo\ i = 1, \ldots, m \quad (14)$$

$$\partial/\partial b[\ L(w, b, \propto)] = -\sum_{i=1}^{m} \propto_i y_i = 0 \quad (15)$$

$$\partial/\partial w[L(w, b, \propto)] = \quad w - \sum_{i=1}^{m} \propto_i y_i \times_i = 0 \quad (16)$$

La parte de la función representada por la expansión (16) trabaja con un pequeño conjunto de datos de entrenamiento conocidos como Vectores de Soporte (SV), sobre los cuales se forman los hiperplanos de separación del clasificador y para los cuales se debe cumplir las condiciones de KKT (Karush-Kuhn-Tucker) (14), en donde los multiplicadores de Lagrange deben ser iguales a cero ($\propto_i = 0$).

Reemplazando (15) y (16) en (13) y bajo el supuesto de una clasificación lineal, se aplica la teoría de multiplicadores Lagrangianos para obtener una función prima objetivo que debe ser Maximizada:

$$\underset{\propto \in \mathbb{R}^m}{\text{Maximizar}} \sum_{i=1}^{m} \propto_i - \frac{1}{2} \sum_{i,j=1}^{m} \propto_i \propto_j y_i y_j K_{ij} \quad \propto \in \mathbb{R}^m \quad (17)$$

Sujeto a:

$$\propto_i \geq 0, \quad para\ todo\ i = 1, \ldots, m \quad y \quad \sum_{i=1}^{m} \propto_i y_i = 0$$

Donde $K_{ij} := \langle x_i x_j \rangle$ representa la función Kernel.

La función de decisión puede escribirse como:

$$f(x) = sgn\left(\sum_{i=1}^{m} y_i \propto_i \langle x_i x_j \rangle + b\right) \tag{18}$$

2.4.3 Clasificación linealmente posible con parámetros de sintonización

Para una clasificador lineal SVM que presente problemas de desempeño ya sea en su capacidad de aprendizaje o en la maximización del margen, se propone la introducción de algunos parámetros de sintonización que sirven para mejorar dicho desempeño. Uno de estos, es una variable de relajación (slack) ξ_i $(\xi_i \geq 0)\, para\ todo\ i = 1, \dots, m$, la cual es incluida en (9), quedando así:

$$y_i.(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) \geq 1 - \xi_i \qquad para\ todos\ los \quad i = 1, \dots, m\ y\ \xi_i \geq 0 \tag{19}$$

Además de la variable de relajación ξ_i, también se incluye una constante C conocida como penalización, quedando entonces una función prima objetivo que debe ser Minimizada:

$$Minimizar\ \ 1/2\|w\|^2 + \mathrm{C}\sum_{i=1}^{m} \xi_i \quad \text{Sujeto a (19)} \tag{20}$$

Con estos parámetros adicionales se quiere encontrar un equilibrio entre maximización del margen y la minimización del error de entrenamiento. Con el primer término de (20) se controla la capacidad de aprendizaje y con el segundo término se controla la rigidez o flexibilidad para aceptar errores por medio de la constante de penalización (C), mientras el término ξ_i determina el número de errores permitidos. Un valor de penalización muy alto (C↑) hará que el clasificador sea muy flexible y permita múltiples errores, por otro lado

(C=0), hará que el clasificador sea demasiado rígido y no admita errores. Existen varios criterios para la selección de los valores para C, sin embargo muchos trabajos recomiendan ajustarlos de acuerdo a las características propias del problema de clasificación.

2.4.4 La función Kernel en SVM no lineales

Las SVM proporcionan un hiperplano óptimo que separa, con un margen máximo, el conjunto de datos si estos son linealmente separables, estos casos se estudiaron en el numeral anterior donde puede existir una clasificación lineal sencilla y una clasificación lineal controlada con variables de relajación ξ_i y una constante de penalización por error de clasificación C. Sin embargo, si los datos en su espacio de representación original no son separables linealmente, aunque se les aplique el tratamiento previamente visto, entonces será necesario implementar una función kernel que cree un espacio de características de mayor dimensión para representar los datos y así garantizar que las clases sean separables, a este clasificador se le conoce también como clasificador de margen suave o blando.

Con la inclusión de ésta función kernel subyacen algunos cuestionamientos: ¿Cómo construir un hiperplano de separación óptimo en un espacio de representación de alta dimensión? ¿Cómo interpretar un espacio de representación de alta dimensión y cómo se representarán los datos ahí?, entre otros. Dado dos patrones $x\ y\ x_i$, lo que se quiere es crear el hiperplano de separación de tal forma que, de los datos no se necesite explícitamente su mapeo, sino que dependa del producto punto entre ellos en el espacio de representación (Bodo, 2008), lo cual dará como resultado un escalar.
Esto es:

$$K(x, x') \coloneqq \langle \times, \times' \rangle = \langle \Phi(x), \Phi(x') \rangle \tag{21}$$

$$\Phi: X \to \mathcal{H}, dónde\ x \mapsto \times \coloneqq \Phi(x) \tag{22}$$

Donde $\mathcal{H}$ es conocido como el espacio de características. Existen dos beneficios importantes como resultado de empotrar los datos en $\mathcal{H}$ vía Φ, primero permiten el tratamiento geométrico con los datos lo cual facilita el estudio y la implementación de algoritmos usando algebra lineal y geometría analítica. Segundo, permite definir una medida de similitud desde el producto punto en $\mathcal{H}$, lo cual proporciona libertad para la selección del mapeo Φ, con lo que se logra una gran flexibilidad y modularidad para el diseño e implementación de medidas de similitud y algoritmos de aprendizaje.

La función (18) se convierte en:

$$f(x) = sgn\left(\sum_{i=1}^{m} y_i \propto_i K_{ij} + b\right) \tag{23}$$

$$f(x) = sgn(\sum_{i=1}^{m} y_i \propto_i \langle \Phi(x), \Phi(x') \rangle + b) \tag{24}$$

Aplicando multiplicadores Lagrangianos, se debe maximizar:

$$\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^{m} \propto_i - \frac{1}{2} \sum_{i,j=1}^{m} \propto_i \propto_j y_i y_j \langle \Phi(x), \Phi(x') \rangle \quad \propto \in \mathbb{R}^m \tag{25}$$

2.4.5 El truco Kernel

La función Kernel hace posible convertir lo que sería un problema de clasificación no lineal en el espacio dimensional original, a un sencillo problema de clasificación lineal en un espacio dimensional mayor a través de transformaciones no lineales de algoritmos lineales basados en el truco kernel sustentado en el Teorema de Mercer.

La transformación no lineal o truco kernel consiste en encontrar una expresión del producto punto del espacio de representación no lineal de entrada. Para entender un poco más este concepto se estudiará una transformación no lineal a un espacio de mayor dimensionalidad de un espacio de características polinómico de dos dimensiones de grado 3 como el que se muestra en la ecuación 26.

(26)

$$\Phi(x) = \left[x_1^3, x_2^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1x_2, \sqrt{3}x_1, \sqrt{3}x_2, 1\right]$$

Al realizar el producto punto entre dos muestras en el espacio de características:

$$\Phi(x)^T\Phi(y) = x_1^3y_1^3 + x_2^3y_2^3 + 3x_2y_2x_1^2y_1^2 + 3x_1y_1x_2^2y_2^2 + 3x_1^2y_1^2 + 3x_2^2y_2^2 + 6x_1y_1x_2y_2 + 3x_1y_1 + 3x_2y_2 + 1$$

$$\Phi(x)^T\Phi(y) = (x_1y_1 + x_2y_2 + 1)^3$$

Observando el resultado, se encuentra una función que describe el producto punto con una complejidad mucho menor, pasando de 10 a solo 3 multiplicaciones, con lo que se obtiene una expresión matemática del producto punto en función del espacio de entrada original, esta expresión se denomina función Kernel, y el espacio de mayor dimensión es un espacio de Hilbert $\mathcal{H}$ (Reproducing Kernel Hilbert Space, RKHS).

La transformación aumenta la posibilidad de que haya una separabilidad lineal de cualquier conjunto de datos, además con una función kernel adecuada, cualquier algoritmo que pueda expresarse en función de productos punto sobre su espacio de entrada puede ser kernelizado, sustituyéndose en el algoritmo el producto escalar "original" por la función kernel, lo que significa que implícitamente el algoritmo "original" pase a aplicarse sobre el espacio de características $\mathcal{H}$. El "truco del kernel" permite que algoritmos "lineales" se apliquen sobre problemas "no lineales".

El Teorema de Mercer permite identificar y caracterizar las funciones kernel, determinando si existe un par $\{\mathcal{H}, \Phi\}$, es decir, un espacio de Hilbert y una Transformación no lineal, con la propiedad de encontrar una expresión matemática del producto punto en función del espacio de entrada original.

Definiciones previas:

$k = X\ x\ X \rightarrow \mathbb{R}$ es simétrica si $k(x,y) = k(y,x)\ \forall x, y\ \in X$

$k = X\ x\ X \rightarrow \mathbb{R}$ es semidefinida positiva si se verifica que $\sum_{i=1}^{n}\sum_{j=1}^{n} c_ic_j\left(x_i, y_j\right) > 0$ para cualquier conjunto de objetos $x_1, x_2, \ldots, x_n$ de X y cualquier conjunto de valores reales $c_1, c_2, \ldots, c_n$.

Teorema de Mercer:

Para cualquier función $k = X\ x\ X \rightarrow \mathbb{R}$ que sea simétrica y semidefinida positiva existe un espacio de Hilbert $\mathcal{H}$ (espacio vectorial de dimensión N con propiedades equivalentes a las de $\mathbb{R}^N$) y una función $\Phi : X \rightarrow \mathcal{H}$ tal que:

$$k(y,x) = \Phi(x).\Phi(y)\ \forall x,y\ \in X \tag{27}$$

2.4.6 6 Tipos de Kernel

Existen diferentes funciones Kernel, siendo las más comunes:

Kernel lineal: $K(x_i, x_j) = x_i^T x_j$

Kernel polinómico: $K(x_i, x_j) = (x_i^T x_j + 1)^n$

Kernel Gausiano: $K(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

Se pueden realizar combinación de kernels. Si consideramos las siguientes premisas:

- $K1\ y\ K2$ son kernels en X.
- $a > 0$.
- $f: X \rightarrow \mathbb{R}$.
- $\emptyset: X \rightarrow \mathbb{R}^m$.
- $K3$ es un kernel en $\mathbb{R}^m$.

Entonces todas las siguientes funciones son kernels en X:

- $K(x_i, x_j) = K1(x_i, x_j) + K2(x_i, x_j)$
- $K(x_i, x_j) = a.K1(x_i, x_j)$
- $K(x_i, x_j) = K1(x_i, x_j)\ .K2(x_i, x_j)$
- $K(x_i, x_j) = f(x_i)\ .f(x_j)$
- $K(x_i, x_j) = K3\left(\emptyset(x_i), \emptyset(x_j)\right)$

2.4.7 Máquinas de Vectores de Soporte Semi-supervisadas (S3VM)

Durante años se han propuesto muchos algoritmos para el aprendizaje semi-supervisado, uno de los más usados son las máquinas de vectores de soporte semi-supervisadas, más conocidas como S³VM, técnica que ha dado origen a diferentes corrientes como se muestra en la Figura 1. Este método de clasificación se basa en la maximización del margen en la frontera de decisión en máquinas de soporte no lineales (Bach, Lanckriet, & Jordan, 2004), es decir, cuando la separación de clases es linealmente imposible, lo que hace necesario la incursión de una función que resuelva el problema computacional de mapear los datos en el espacio de entrada y crear una dimensión de representación mucho más grande para construir en esta un hiperplano de separación, para lo que se determinó que el mapeo de los datos no requería ser explícito en el algoritmo de entrenamiento, basta con obtener una función de mapeo no lineal llamada Kernel, la cual realiza el producto punto entre los datos en dicho espacio.

Sin embargo, a pesar de que estas técnicas han mostrado ser eficientes en diferentes campos, existe un problema inherente a su utilización: la selección del Kernel en el clasificador S³VM depende de la presunción geométrica que se haga sobre los datos. La elección incorrecta del Kernel para un grupo de datos, producirá un desempeño pobre en el clasificador (Dai & Yeung, 2007). Según (Baghshah, 2010) los conjuntos de datos que se obtienen en el mundo real puede contener grupos con diferentes formas, tamaños, escasez de datos, y diferentes grados de separación. Por lo tanto, los métodos de agrupamiento (clustering) requieren la definición de una medida de similitud adecuada entre los patrones.

El método de agrupación afecta críticamente la elección de una función kernel apropiada, por tal razón en los últimos años ha habido un creciente interés en el aprendizaje de métricas para la configuración de aprendizajes semi-supervisados (en particular, para agrupaciones semi-supervisadas), por lo que se ha introducido los parámetros de limitación conocidos como similitud (musk-link) y disimilitud (cannot-link), como un tipo popular de información lateral usada para disminuir así las restricciones de los métodos usados tradicionalmente.

Problema de programación cuadrática

Si consideramos un problema de programación no lineal, cuya función objetivo es la suma de términos de la forma $X_1^{n_1} X_2^{n_2} \ldots\ldots X_n^{n_n}$, el grado del término es $n_{1+} n_2 + \cdots + n_n$, se presenta un problema de programación no lineal, cuyas restricciones son lineales y cuya función objetivo es la suma de términos de la forma $X_1^{n_1} X_2^{n_2} \ldots\ldots X_n^{n_n}$ (en la cual cada término tiene un grado de 2, 1 o 0). Esto es lo que se denomina un problema de programación cuadrática. Existen algoritmos con los que se optimiza el problema de programación cuadrática QP (Quadratic Programming), como el SMO (Sequential Minimal Optimization) presentado por (Platt, 1999).

El entrenamiento de una SVM requiere la solución de un gran problema de optimización de QP. El algoritmo SMO separa este gran problema QP en una serie de posibles QP mucho más pequeños, los cuales son resueltos analíticamente minimizando el tiempo de computación interna, además la cantidad de memoria requerida es lineal en el set de entrenamiento, lo que a su vez permite manejar grandes conjuntos de datos de entrenamiento.

$$\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^{m} \propto_i - \frac{1}{2} \sum_{i,j=1}^{m} \propto_i \propto_j y_i y_j \langle \Phi(x), \Phi(x') \rangle \quad \propto \in \mathbb{R}^m$$

$$0 \leq \propto_i \leq C, \forall i,$$

$$\sum_{i=1}^{m} y_i \propto_i = 0$$

El problema QP es resuelto cuando, para todo i:

$$\propto_i = 0 \rightarrow y_i f(\overrightarrow{x_i}) \geq 1$$

$$0 \leq \propto_i \leq C \rightarrow y_i f(\overrightarrow{x_i}) = 1$$

$$\propto_i = C \rightarrow y_i f(\overrightarrow{x_i}) \leq 1$$

El algoritmo SMO resuelve el problema QP cumple con las condiciones KKT y puede ser evaluado en cualquier momento.

2.5 Métricas de distancia

En bioinformática, el establecimiento de clústers que reúnen secuencias protéicas derivadas de un mismo árbol evolutivo, crea la necesidad de introducir los conceptos de agrupación de genes de acuerdo a la distancia que existe entre ellos. Al medir una distancia entre dos secuencias de genes similares, se concluye que éstas realizan la misma función biológica en el árbol evolutivo, cuando se cumple con el puntaje mínimo para pertenecer a un mismo árbol filogenético.

Las medidas de similitud entre dos secuencias pueden determinarse de forma geométrica a través de la inclusión de una función de distancia que puede ser lineal, polinomial, gaussiana, entre otras; pero también puede realizarse a través del análisis de kernels de secuencias como Spectrum, Mismatch, Gappy, en donde se comparan las cadenas de elementos que la componen.

2.5.1 Métricas de distancia geométricas

Existen diferentes funciones métricas de distancia, y la selección de una de ellas depende de las características propias del conjunto de datos y de su dimensión, lo cual afecta considerablemente la correlación entre ellos, provocando a su vez errores en los resultados de clasificación.

Asumiendo que se tiene un conjunto S de elementos y una función d, la cual para todos los pares de elementos ordenados (a, b) del conjunto S, retorna la distancia $d(a, b)$ desde a hasta b. Algunas propiedades de la función d son:

- $\forall\, a, b \in S : d(a, b) = d(b, a)$ (Simetría)
- $\forall\, a \in S : d(a, a) = 0$ (Marca de identificación)
- $\forall\, a, b \in S : d(a, b) = 0\ Si\ y\ solo\ si\ a = b$ (Definida)
- $\forall\, a, b \in S : d(a, b) \geq 0$ (No negativa)
- $\forall\, a, b, c \in S : d(a, b) \leq d(a, c) + d(c, b)$ (Desigualdad triangular)

Para realizar la clasificación en sistemas de múltiples dimensiones es necesario calcular la distancia entre vectores de características, utilizando métricas de distancia que cumplan con las propiedades antes descritas.

Se han realizado diferentes trabajos donde se estudia la importancia de la métrica seleccionada. Aunque distancias como la Euclídea y la Manhattan han sido propuestas en la literatura para medir similitud entre vectores de características (Vadivel, A., Majumdar, A. & Sural, S., 2003), también se han explorado otras medidas de similitud para encontrar la mejor métrica de distancia en aplicaciones como recuperación de la imagen de textura convencional (Kokare, M., Chatterji, B. N., & Biswas, P. K., 2003), allí además de las métricas convencionales, se realiza una comparación de medidas de similitud con métricas como Mahalanobis, Chebychev, Canberra, Chi Cuadrado, entre otras.

Para dos vectores $\boldsymbol{a}$ y $\boldsymbol{b}$ de múltiples dimensiones, donde $\boldsymbol{a} = (a_1, a_2, a_3, \dots, a_n)$ y $\boldsymbol{b} = (b_1, b_2, b_3, \dots, b_n)$, existen algunas métricas de distancia conocidas que sirven para determinar la similitud que existe entre ellos.

Distancia Euclídea

La distancia Euclídea, conocida también como $L2$, utiliza el Teorema de Pitágoras para calcular la distancia entre dos vectores cualesquiera. Esta métrica se basa en un espacio

Euclídeo, que a su vez es un espacio vectorial dotado de un producto escalar. La distancia Euclídea generalizada es:

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{28}$$

Distancia Mahalanobis

La distancia euclídea normalizada $d(a,b)$ es:

$$d(a,b) = \sqrt{\sum_{i=1}^{n}\frac{(a_i - b_i)^2}{\sigma^2}}$$

En esta métrica de distancia se tienen en cuenta las correlaciones y las varianzas entre los datos a y b, donde σ^2 es la varianza (Cuadras, 1989). Sea $\sum$ la matriz de varianzas-covarianzas, la distancia Mahalanobis se calcula así:

$$d(a,b) = \sqrt{(a_i - b_i)'\sum^{-1}(a_i - b_i)}$$

2.5.2 Métricas basadas en distancias biológicas

Una secuencia de proteínas es una secuencia $X = (x_1, x_2, \dots, x_n)$ sobre el alfabeto de 20 aminoácidos. La medida de similitud / distancia en el conjunto de 20 aminoácidos se define con base en características como longitud de las secuencias, el peso molecular, la polaridad, el punto isoeléctrico y el índice de hidropaticidad.

Para determinar la similitud entre dos secuencias de proteínas se parte de comparar su estructura primaria, la cual está compuesta por un conjunto de aminoácidos dentro del alfabeto de 20 aminoácidos existentes. Esta comparación se puede hacer de manera manual o aplicando algoritmos computacionales, lo cual dependerá de las longitudes de las mismas. A este método se le llama alineamiento de secuencias proteicas, y lo que

busca es identificar si una cadena de aminoácidos de una secuencia, comparte propiedades químicas similares con una cadena de aminoácidos de otra, a través del análisis de la cadena lateral de los aminoácidos, así se podrá identificar si dos secuencias pertenecen a un mismo patrón o linaje, es decir, si cumplen el mismo propósito biológico, entre los que están: procesos biológicos asociados, componentes celulares o funciones moleculares en especies.

Existen algoritmos que cuantifican el parecido entre diferentes aminoácidos de acuerdo al número de veces que suelen aparecer en la misma posición en proteínas homólogas, con lo que se puede determinar si son funcionalmente equivalentes. Una de las técnicas consiste en matrices donde se compara cada uno de los aminoácidos con todos los demás, a cada comparación se le da una puntuación que indica la frecuencia con la que un aminoácido es sustituido por otro en proteínas homólogas. Las dos matrices de puntuación más usadas para calificar el alineamiento de secuencias, son la PAM (Percent Accepted Mutation) y la BLOSUM (BLOcks of Amino Acid SUbstitution Matrix, o matriz de sustitución de bloques de aminoácidos). Estás matrices son utilizadas por algoritmos de alineamiento de secuencias como BLAST (Basic Local Alignment Search Tool) y PSI-BLAST (Position-Specific Iterative-Basic Local Alignment Search Tool), en los que de manera heurística se determina si dos secuencias son homólogas (Henikoff & Henikoff, 1992).

El algoritmo BLAST permite realizar una búsqueda preliminar de similitud entre una secuencia problema (query) y las bases de datos disponibles en NCBI (National Center for Biotechnology Information), donde fue desarrollado (Schäffer et al, 2001). Ofrece como resultado una medida cuantitativa de la similaridad de la secuencia problema con respecto a las secuencias de las bases de datos, haciendo un alineamiento local por pares de la secuencia problema con cada una de las secuencias de la base de datos con las que muestra alta similitud.

Dentro de las desventajas de estas herramientas computacionales están el no poder hacer búsqueda masiva de similitudes entre secuencias debido a que es un recurso compartido, no se puede personalizar las bases de datos, y las secuencias son enviadas al servidor del NCBI sin ningún tipo de cifrado. Para el caso puntual de la clasificación del conjunto extraído de Embryophyta, cabe anotar que debido a las altas dimensiones del número de

muestras, como del número de características, el uso de estos algoritmos no es el más indicado.

La complejidad para comparar secuencias de proteínas está en que se valoran positivamente también los parecidos de aminoácidos que no son idénticos, pero que son químicamente similares, lo que conduce a que los alineamientos de las secuencias al azar puedan confundirse con alineamientos realmente significativos. Por otro lado cabe anotar que otra de las complejidades en la clasificación de secuencias de proteínas radica en el hecho de que una misma secuencia puede pertenecer simultáneamente a varias funciones biológicas.

2.6 Kernels de secuencias

En la detección de secuencias homólogas usando técnicas de comparación entre los aminoácidos que componen dichas secuencias, se han desarrollado métodos que consisten en determinar si cadenas de aminoácidos o sub-secuencias, son similares entre ellas. A los kernels que se originan de este tipo de comparaciones, se les denomina kernels de secuencias (Hernández, 2013).

Sea el alfabeto Σ, el conjunto de 20 aminoácidos o símbolos que componen las proteínas, y una cadena una secuencia finita de símbolos $s = s_1, s_2, \dots, s_n$ que pertenecen al alfabeto Σ con una longitud l. Existen subsecuencias de s dadas por $i = (i_1, i_2, \dots, i_n)$ que forman la secuencia proteica.

Ejemplo:

Sea $\Sigma = \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ el alfabeto de 20 aminoácidos, y la secuencia proteica $s = AABDCQEGGI$, entonces:

$$s(3) = B$$
$$s(1{:}5) = AABDC$$
$$s(1,2,3,9,10) = AABGI$$

$$l(s) = 10$$

2.6.1 Spectrum kernel

Con el Spectrum kernel (Leslie, C. S., et al., 2002) se comparan dos secuencias, de acuerdo al número de subsecuencias de longitud k que tienen en común. Lo que se hace es transformar el espacio de representación haciendo un mapeo de todas las posibles subsecuencias de entrada en un espacio R^{l^k}, que está dado por:

$$\Phi_k(x) = (\Phi_a(x))_{a \in \sum^k}$$

Donde $\phi_a(x)$= número de veces que aparece el k-mer a en la secuencia x.

Lo que debe definirse para el Spectrum Kernel, es la longitud k de las subsecuencias que componen las dos cadenas, posteriormente se extraen de las dos cadenas las posibles cadenas de longitud k que sean contiguas dentro de cada secuencia y se conocen como k-mers, posteriormente se crea un vector que contendrá el número de veces que aparece cada k-mer dentro de la secuencia. Por último, se calcula el producto punto entre ambos vectores. El kernel entre dos secuencias x y y está definido como:

$$K_k(x, y) = \langle \Phi_k(x), \Phi_k(x) \rangle$$

2.6.2 Mismatch Kernel

El Mismatch kernel propuesto en (Eskin, E., et al., 2002) es muy similar al Spectrum kernel, la diferencia está en la introducción de un concepto biológicamente importante que es la disparidad, esto se hace por medio de un nuevo parámetro llamado m, este parámetro indica el número de disparidades entre cada uno de los k-mer de la secuencia y los k-mer extraídos, es decir, si entre dos k-mer existe un número de disparidades menor a m se

podría decir que ambas sub-secuencias son iguales, para representar el número de disparidades entre dos subsecuencias u y v se utiliza la expresión d(u,v), que simboliza el número de caracteres en que difieren u y v, esta medida es conocida como distancia Hamming (Kuksa, P. P., et al, 2008).

Para el mismatch kernel se realiza un mapeo parecido al mapeo realizado en el Spectrum kernel con una pequeña diferencia y es la inclusión de un parámetro m, que permite un número máximo de disparidades entre los k-mer, cuando m es igual a cero se obtiene el Spectrum kernel, por lo tanto el mapeo queda definido por:

$$\Phi_{k,m}(x) = (\Phi_a(x))_{a \in \sum^k}$$

Existen diversas formas de implementar el mismatch kernel, una de ellas se describe en (Kuksa, P. P., et al, 2008) donde se plantea un algoritmo basado en la distancia Hamming, para ello se precalcula el tamaño de las intersecciones entre los k-mer de las secuencias. El enfoque tradicional usado en (Eskin, E.,et al, 2002) consiste en construir un árbol de profundidad k, donde cada uno de los nodos contiene l ramas , donde l es el tamaño del alfabeto (para el caso de aminoácidos l=20) y cada rama es etiquetada con un símbolo perteneciente al alfabeto. No obstante, en la implementación del algoritmo no es necesario almacenar todo el árbol, la manera de hacerlo es mediante una función recursiva.

2.6.3 Gappy Kernel

El Gappy kernel (Kuksa, P., et al, 2009) tiene dos parámetros k y m, y tiene como objetivo buscar pares de k-mers separados cierta distancia dentro de una secuencia. El parámetro m tiene diferente significado que en el mismatch kernel, ya que, este parámetro representa la distancia a la que pueden estar separados dos k-mer. Además tiene en consideración diferentes mutaciones o transformaciones biológicas tales como inserciones o supresiones. Este kernel no solo toma en consideración la frecuencia en que se encuentra un k-mer dentro de una secuencia, sino que además tienen en cuenta como es la ubicación de los k-mer. La inclusión de información espacial ha demostrado buen desempeño incluso

para subsecuencias pequeñas, por ejemplo k=1 (Kuksa, P., et al, 2009). El Gappy Pair kernel está dado por la siguiente expresión

$$k(s,t) = \sum_{\substack{(a_1,\, k\,, a_2) \\ a_1,\ a_2\ \in\ \sum^k}} C\left(a_1,\, k, a_2\ |s\right) \cdot C(a_1,\, k, a_2\ |t)$$

Donde $C\left(a_1,\, k, a_2\ |s\right)$, representa el número de veces que aparece el substring a_1 separado de a_2 en k caracteres, dentro de la secuencia s.

El tamaño del espacio de características está dado por $|M| = (m+1)\ |\sum|^{2k}$, el cual es considerablemente más grande que para el caso del Mismatch y Spectrum kernel con los mismos valores de k.

2.7 Neighborhood Kernels

Neigborhood Kernels (Weston et al., 2004, 2005; Leslie et al.) es una herramienta de aprendizaje de máquina que se usará para la anotación funcional de proteínas pertenecientes a organismos vegetales. Este algoritmo será utilizado para la construcción de un clasificador con kernel semi-supervisado, su función principal es cambiar la métrica de distancia de tal manera que la distancia relativa entre dos puntos es mucho más pequeña si los puntos están en el mismo clúster, lo cual se logra haciendo un cambio del espacio de representación del clasificador, teniendo en cuenta la estructura descrita por los datos no etiquetados.

El Neighborhood kernel utiliza un promedio de más de un vecino de secuencias definidas por una medida de similitud de secuencia locales, esta técnica semi-supervisada mejora el rendimiento de clasificación cuando se utiliza con kernels de secuencias, logrando

resultados iguales o superiores a los métodos kernel clúster. Por otra parte, este enfoque es mucho más eficiente computacionalmente que otros métodos.

En esta técnica se define una medida de similitud de secuencias estándar neighborhood Nbd (x) para cada secuencia de entrada x, como el conjunto de secuencias x' con score de similitud para x por debajo de un umbral de E-valor fijo, junto con el mismo x. Dado una representación de características de origen fijo, se representa a x por el promedio de los vectores de características para los miembros de su vecindario o neighborhood.

$$\Phi_{nbd}(x) = \frac{1}{|Nbd(x)|}\sum x' \in Nbd(x)\Phi\mathrm{orig}(x')$$

Se define el neighborhood kernel por:

$$K_{nbd}(x,y) = \frac{1}{|Nbd(x)||Nbd(y)|}\sum_{x'\in Nbd(x),y'\in Nbd(y)} K_{oring}(x',y')$$

2.8 Balanceo de clases usando W-SVM

Existen problemas para la clasificación cuando el desbalance entre las clases es predominante, es decir, cuando existen muy pocas muestras de una clase, y un elevado número de muestras de la otra. La mayoría de los clasificadores son incapaces de establecer cuál es la diferencia entre el costo de los falsos positivos y los falsos negativos (Thai-Nghe, N., et al., 2010), y aunque existen problemas de clasificación en los que no es relevante asignarle costos diferentes a las clases, en el problema de clasificación de secuencias proteicas que se quiere resolver, sí lo es, porque debido a la gran diferencia que existe entre el número de muestras que pertenecen a una clase u otra, a la alta dimensionalidad de los datos y sus características, y al gran número de muestras sin etiqueta, es importante proveer al clasificador de una herramienta que le permita un aprendizaje basado en criterios de sensibilidad de costo extraídos de la poca información conocida que se tienen sobre la base de datos, así se podrán usar todas las muestras en

la etapa de entrenamiento, garantizando que haya representación de ambas clases. Recordando La función prima objetivo de una SVM:

$$Minimizar\ \ 1/2\|w\|^2 + \mathrm{C}\sum_{i=1}^{m}\xi_i$$

El Clasificador W-SVM (Weighted-SVM) (Yang, Song & Wang, 2007), asigna un peso diferente a cada clase por medio del coeficiente de regularización $\boldsymbol{C}$ de la SVM y $\boldsymbol{W_i}$. En el caso de un clasificador binario, serían los pesos $\boldsymbol{W_1}\ \boldsymbol{y}\ \boldsymbol{W_2}$.

$$Minimizar\ \ 1/2\|w\|^2 + \mathrm{C}\sum_{i=1}^{m}W_i\xi_i$$

2.9 Optimización por enjambre de partículas PSO

Este método evolutivo propuesto por (Kennedy & Eberhart, 1995) está inspirado en el comportamiento de bandadas de aves o cardumen. Con él se busca llegar a la solución óptima entre varias opciones o partículas de un enjambre, así, las variables de ubicación, velocidad y aceleración de cada partícula, son actualizadas a medida que se intercambia información entre las partículas, hasta encontrar los mejores candidatos, proceso condicionado por dos aspectos, uno de ellos, por el cumplimiento de una función aptitud establecida, y el otro, que tiene que ver con los criterios de parada, que pueden ser por el número de iteraciones o estableciendo un umbral de error aceptable.

Una partícula o solución es representada en PSO por un vector. Cada partícula i tiene una dimensión n y se representa:

$$X(i) = (x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)})$$

La velocidad o razón de cambio de cada partícula es:

$$V(i) = (v_{(i,1)}, v_{(i,2)}, \dots, v_{(i,n)})$$

Las partículas vuelan por el espacio de alta dimensión buscando la posición óptima, la cual es almacenada en memoria, así:

$$P(i) = (p_{(i,1)}, p_{(i,2)}, \dots, p_{(i,n)})$$

La mejor partícula del enjambre es:

$$G(i) = (g_{(i,1)}, g_{(i,2)}, \dots, g_{(i,n)})$$

Se debe actualizar la posición y la velocidad de cada partícula teniendo en cuenta su propia inercia w y de acuerdo al grado de confianza (en el rango 0-1) que se establece por las variables aleatorias $r_1 \, y \, r_2$. Las constantes de aceleración $c_1 \, y \, c_2$, llamadas también coeficientes de confianza, los cuales se encargan de dirigir a las partículas hacia las posiciones $P \, y \, G$ (Eberhart & Shi, 2001).
La ecuación que describe la velocidad y la posición de cada partícula es:

$$V_{id} = w \times V_i + \, c_1 \times r_1 \times (pbest_{id} - X_{id}) + \, c_2 \times r_2 \times (gbest_d - \, X_{id})$$

En esta ecuación el término id hace referencia a la partícula i en la dth dimensión, $Pbest$ Es el valor de la mejor posición de la partícula, $lbest$ son las mejores localizaciones encontradas por otras partículas a medida que recorren el espacio de búsqueda y $gbest$ es el mejor valor global o posición global que ha sido visitado por todas las partículas.

2.10 Validación cruzada

Es una técnica usada para validar estadísticamente los resultados de un modelo de predicción propuesto, de tal manera que se garantice una independencia entre las particiones realizadas a los datos, es decir, entre los datos de entrenamiento y los datos de prueba. Cross-validation, como es conocida, consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones realizadas a los datos. Esta técnica es muy usada para entornos donde el objetivo principal es la predicción y se requiere estimar la precisión del modelo de predicción. Finalmente esta técnica ofrece información acerca del error generalizado o error total cometido en la clasificación de los datos (Duan, K., et al., 2003).

2.11 Medida de desempeño

Existen varias medidas de desempeño para estimar el error en un problema de clasificación, sin embargo, debido al desbalance que existe entre las clases, algunas medidas de desempeño no representan de forma correcta el desempeño global de un clasificador. Para este trabajo se evaluará el desempeño de la SVM mediante una matriz de confusión como se recomienda en (Chawla, N. V., et al, 2002) para evaluar el desempeño de un algoritmo de aprendizaje. La matriz de confusión se representa en la Tabla 3, así:

Tabla 4: Matriz de confusión.

Con esta matriz se calcula la tasa de verdaderos positivos y la tasa de verdaderos negativos, para calcular posteriormente la especificidad y la sensibilidad del modelo predictivo.

		Clase Estimada	
		-	+
Clase Real	-	VN	FP
	+	FN	VP

Donde,

VN es el número de verdaderos negativos e indica el número de muestras de la clase negativa que fueron clasificadas correctamente.

VP es el número de verdaderos positivos e indica el número de muestras de la clase positiva que fueron clasificadas correctamente.

FN es el número de falsos negativos e indica el número de muestras de la clase negativa que fueron clasificadas incorrectamente.

FP es el número de falsos positivos e indica el número de muestras de la clase positiva que fueron clasificadas incorrectamente.

Para medir el desempeño global en un problema de clasificación, se han creado algunas medidas que estiman de forma adecuada el error en problemas de clasificación que involucran desbalance de clases, estas medidas son creadas a partir de la sensibilidad y la especificidad; también conocidas como tasa de verdaderos positivos y tasa de verdaderos negativos, respectivamente.

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

$$Especificidad = \frac{VF}{VN + FP}$$

Una medida utilizada para estimar el desempeño de un clasificador es la media geométrica entre la sensibilidad y la especificidad; esta medida representa el balance entre las muestras bien clasificadas de la clase positiva y las muestras bien clasificadas de la clase negativa (Tang, Y., et al., 2010). La media geométrica está definida como la raíz n-ésima del producto de n valores.

Si se quiere calcular la media geométrica entre la sensibilidad y la especificidad, se utiliza la siguiente expresión:

$$G(x_1, \dots, x_n) = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

Si se quiere calcular la media geométrica entre la sensibilidad y la especificidad, se utiliza la siguiente expresión:

$$G(sensibilidad, especificidad) = \sqrt{sensibilidad * especificidad}$$

3. Marco Experimental

Para el desarrollo de la metodología, primero se realizará la construcción de un espacio de representación que permita el tratamiento estadístico de los datos obtenidos mediante la extracción de características físico-químicas hecho a secuencias proteicas de organismos vegetales. Posteriormente se aplicarán diferentes métricas de distancia geométricas como la Euclídea y la Mahalanobis, así como métricas de similitud de secuencias usando kernels de secuencias como Mismatch y Gappy, en la construcción del clasificador SVM semi-supervisado basado en Neighborhood Kernels. Por último, se validará el modelo sobre un conjunto de datos de secuencias proteicas y se evaluará su desempeño usando validación cruzada.

3.1 Base de datos

El proyecto GO (Ontología de Genes) es un esfuerzo colaborativo que surge en 1998, como necesidad de descripciones consistentes de genes en las bases de datos. El consorcio GO (GOC) ha crecido considerablemente desde entonces, incorporando a sus bases de datos repositorios para plantas, animales y otros genomas. El proyecto GO se encarga de construir un vocabulario controlado y estructurado, conocido como ontologías, que se usa para ser aplicado en la anotación de secuencias proteicas, genes, o productos de genes, en bases de datos biológicas (The Gene Ontology consortium, 2004). Existen tres ontologías que describen los genes en términos de sus procesos biológicos asociados, componentes celulares y funciones moleculares en especies, cada una con un conjunto de etiquetas o “GO-terms”.

La base de datos Embryophyta está disponible en UniProtKB/Swiss-Prot (Jain et al., 2009). Ésta base de datos fue depurada en (Jaramillo-Garzón et al., 2013) para obtener un

conjunto de datos de proteínas correspondientes a plantas terrestres que incluye árboles, flores, helechos, musgos y otros tipos de plantas terrestres verdes, para un total de 8879 muestras de secuencias proteicas que pueden pertenecer a una o varias de las ontologías mencionadas, simultáneamente. Ésta base de datos fue filtrada usando el software CD-HIT (Li & Godzik, 2006), con el que se analizaron, compararon y agruparon las secuencias proteicas obteniendo un total de 3368 secuencias, con una similitud de tan solo el 30% entre ellas, lo que hace que la predicción de funciones sea una tarea compleja que debe valerse de algoritmos robustos que sean capaces de usar esa baja similitud y traducirla en un porcentaje de acierto importante, sobre todo porque existe un desbalance considerable entre las muestras y una gran cantidad de falsos positivos en el conjunto de datos (Rhee et al., 2008). Adicionalmente se utilizó una base de datos, también extraída del conjunto original Embryophyta, con 26226 secuencias proteicas sin etiquetar, es decir, secuencias de las que no se conoce con certeza su correspondencia a alguno de los "GO-terms" existentes en las diferentes ontologías, y cuyo propósito en este trabajo, es ayudar al entrenamiento de la máquina para clasificar las secuencias con similitud del 30%, mediante el reconocimiento de patrones usando técnicas de aprendizaje de máquina semisupervisado, donde intervienen muestras etiquetadas y muestras sin etiquetar.

Después de analizar las secuencias proteicas y los GO-terms a los que éstas pertenecían, se determinó descartar aquellos GO-terms que tuviesen menos de 30 proteínas, porque no son un número representativo de muestras para realizar el entrenamiento de la máquina, por lo tanto los resultados que se obtendrían no serían estadísticamente confiables (Jaramillo-Garzón et al., 2013).

3.2 Extracción de características

El conjunto de datos etiquetados se compone de 3368 muestras o secuencias proteicas, cada secuencia con un total de 544 características propias de los aminoácidos. Se utilizó la base de datos "aa.index", que contiene índices o valores numéricos que representan alguna de las diferentes propiedades físico-químicas y biológicas de los 20 aminoácidos presentes en las proteínas. Esta base de datos, que contiene 544 índices, fue computada

con la base de datos de secuencias de proteínas aplicando un algoritmo de extracción de características, con lo que se redujo a 438 el número de características relevantes clasificadas en tres tipos de atributos, como son: las características físico-químicas, la composición de la primera estructura y la composición de la segunda estructura de las proteínas. Dentro de las propiedades físico-químicas están la longitud de las secuencias, el peso molecular, la polaridad, para cada aminoácido se calcula el punto isoeléctrico, y por último, el índice de hidropaticidad GRAVY (índice hidrofílico e hidrofóbico de la cadena lateral R), el porcentaje de existencia de cada aminoácido en la secuencia, el porcentaje de existencia de cada dímero en una secuencia, porcentaje de aminoácidos cargados positiva y negativamente en las secuencias. El segundo grupo de características se hizo con la estructura primaria de la secuencia proteica, donde se determinó la frecuencia de aparición de cada aminoácido en la misma, como también la frecuencia de aparición de arreglos pares de aminoácidos o dímeros, evitando arreglos con un número mayor debido a que el espacio de características crecería exponencialmente con el mismo. El tercer atributo está orientado a la predicción de la estructura secundaria de las proteínas, tal predicción fue realizada con el software Predator 2.1 (Frishman & Argos, 1997) con el que se generaron secuencias secundarias (2D) lineales en tres dominios estructurales: Hélice-α, Hoja-β, y espiral, también se calculó el porcentaje de cada estructura secundaria en la secuencia.

Lo primero que se hizo fue tomar las 3368 secuencias proteicas y asignarles un conjunto de símbolos o caracteres al conjunto de aminoácidos que las componen según la Tabla 1, posteriormente se crean vectores de probabilidad donde se computa cada conjunto de aminoácidos que componen las secuencias, con su probabilidad de aparición en la naturaleza, según (Buxbaum, 2007).

En la Tabla 4 se observa el conjunto de características extraído de cada secuencia de aminoácidos, divididos en tres grandes atributos: físico-químicos, estructura primaria y estructura secundaria, también se presenta la descripción del conjunto de características correspondientes a cada tipo.

Tabla 5. Conjunto de características extraída a la secuencia de aminoácidos

Atributo	Característica	Número
Físico-químicas	Longitud de la secuencia	1
	Peso molecular	1
	Cargados positivamente	1
	Cargados negativamente	1
	Punto isoeléctrico	1
	Índice de hidropaticidad (GRAVY)	1
Estructura primaria	Frecuencia de aminoácidos	20
	Frecuencia de dímeros de aminoácidos	400
Estructura secundaria	Frecuencia de las estructuras 2D.	3
	Frecuencias de dímeros de estructuras 2D.	9
Total		**438**

3.3 Selección de Características

Debido a la alta dimensión de los datos, y al conjunto de características, se pensó en utilizar una técnica de selección que permita reducir el conjunto de características de una manera eficiente y efectiva, partiendo del análisis de dos características en el conjunto de datos, como son la relevancia y la redundancia. Una relevancia fuerte de una característica indica que ésta es siempre necesaria para un subconjunto óptimo, por lo tanto no puede ser removida sin que se afecte la distribución original de clases, por su parte, una característica es débilmente relevante si ésta no es siempre necesaria, pero puede llegar a serlo para un subconjunto bajo ciertas condiciones; por último, es irrelevante si, al ser eliminadas, no afecta la distribución original de los datos.

La redundancia por su parte, se define en términos de correlación, es decir, dos muestras son mutuamente redundantes si sus valores están completamente correlacionados. Existen dos enfoques en la selección de características, uno donde se evalúa individualmente las muestras, y otro donde se evalúa subconjuntos de muestras. Cuando se trata de datos de alta dimensión se aplica el segundo método. Para determinar la

redundancia entre una muestra y un subconjunto de datos, se debe realizar una estrategia de selección de características avanzada, orientada a evaluar la relevancia y la redundancia entre subconjuntos de datos generados hasta encontrar aquellos óptimos que satisfagan ciertos criterios. El algoritmo de análisis de relevancia y redundancia utilizado para la selección de características, fue el propuesto en (Yu & Liu, 2004) denominado FCBF (Fast Correlation-Based Filter). En este algoritmo primero se selecciona un subconjunto de muestras relevantes, y posteriormente se seleccionan del mismo, las más predominantes.

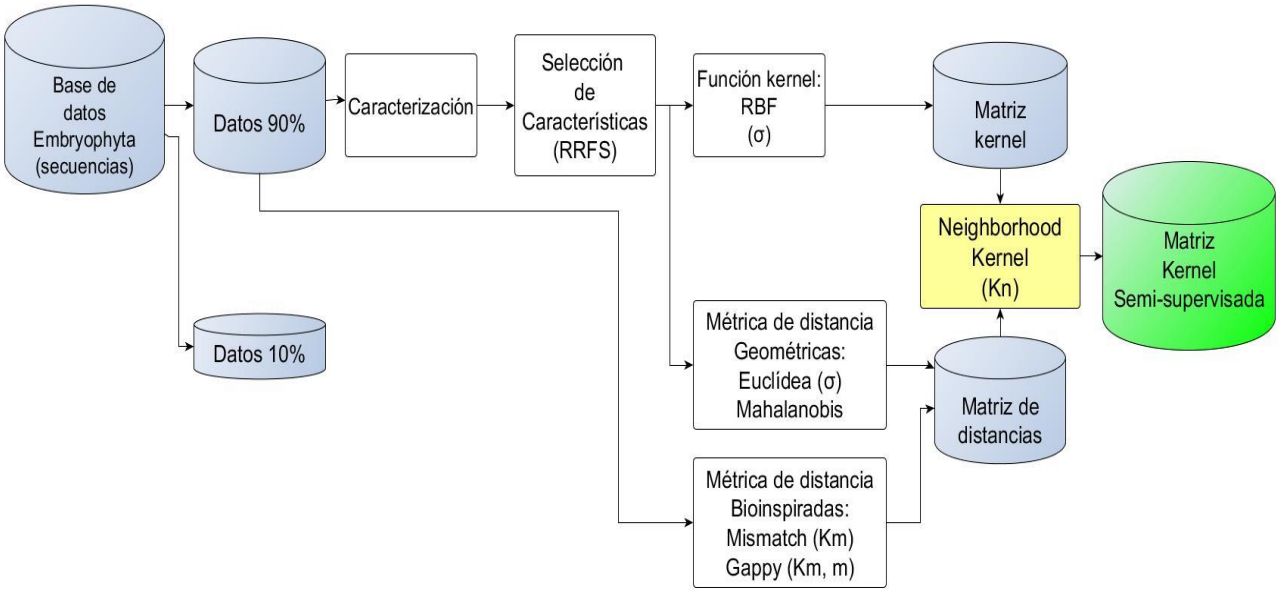
3.4 Construcción de Matriz Kernel semi-supervisada

La matriz kernel semi-supervisada se forma a partir de la base de datos Embryophyta, de la cual se selecciona el 90% de los datos de manera aleatoria, a los que se les extrae y selecciona las características más relevantes. Una vez obtenidas las matrices de los 14 problemas de clasificación con sus dimensiones reducidas, se genera una matriz kernel con función de distancia de base radial (RBF).

Por otro lado, se construye una matriz de distancias a la cual se le introduce una métrica de distancia o similitud, que puede ser geométrica o bioinspirada. Las métricas de distancia geométricas escogidas, fueron Euclídea y Mahalanobis, y las medidas de similitud bioinspiradas fueron Mismatch y Gappy. De acuerdo a la métrica de distancia seleccionada, se tienen dos caminos para generar la matriz de distancias, así, en el caso de que sea geométrica, los datos deben tomarse después de haberse realizado la extracción y selección de características, por el contrario, si la métrica es bioinspirada, los datos se toman directamente del 90% de los datos seleccionados de la base de datos principal.

Una vez obtenidas las matrices kernel y de distancias, se introducen en el algoritmo Neighborhood Kernel para que éste genere la matriz kernel semi-supervisada como se muestra en la Figura 9.

Figura 9. Matriz Kernel Semi-supervisada



Los algoritmos usados para formar las matrices kernel y de distancias requieren el ajuste de unos parámetros de sintonización que permitan que las matrices resultantes mejoren el desempeño del clasificador. La función Kernel Gaussiana RBF tiene como parámetro de sintonización el σ (sigma), el cual determina la medida de distancia entre dos datos evaluados. La función kernel RBF se define así:

$$\text{Kernel Gausiano: } K(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Los parámetros que caracterizan a la matriz de distancias, dependen del tipo de métrica seleccionada, así, para las dos medidas geométricas, la Euclídea y la Mahalanobis, se tienen en cuenta σ y Σ^{-1} (sigma y matriz de varianzas-covarianzas) entre los datos.

Por su parte, en las medidas de similitud biológicas, los parámetros a ajustar serán, K_m que representa el número sub-secuencias de longitud k que sean contiguas dentro de la secuencia (k-mers en la literatura), para la medida Mismatch. Además de K_m, la medida

Gappy introduce el parámetro m, que representa la separación que puede haber entre dos pares de k-mers en una secuencia.

3.5 Sintonización de los parámetros del modelo

En la metodología propuesta (Figura 10) se observan los parámetros de sintonización del modelo escritos entre paréntesis, llamados también hiper-parámetros. Se utilizaron dos algoritmos de sintonización de parámetros, W-SVM (Yang, Song & Wang, 2007) y PSO (Kennedy & Eberhart, 1995). El primero se utilizó para el problema de desbalance entre las clases. En la Tabla 5 se observa el número de muestras perteneciente a cada una de las funciones, entre un total de 3368 muestras. Por ejemplo, para la función NtBind se tienen 47 muestras de la clase positiva, mientras que 1506 pertenecen a la clase negativa, el resto son muestras que no están etiquetadas.

Tabla 6. Tamaño de las clases

Función molecular	Acrónimo	Tamaño clase positiva	Tamaño clase negativa
Nucleotide binding	Ntbind	47	1506
Molecular function*	MF*	268	1705
DNA binding	DnaBind	107	1430
Transcription factor activity	TranscFact	307	1402
RNA binding	RnaBind	43	1493
Catalytic activity*	Catal*	334	1372
Receptor binding	RecBind	38	943
Transporter activity	Transp	125	1583
Binding*	Bind*	173	1534
Protein binding*	ProtBind*	630	968

Kinase activity	Kinase	68	1147
Transferase activity*	Transf*	173	1204
Hydrolase activity	Hydrol	190	1193
Enzyme regulator activity	EnzReg	41	1665

Debido a que el desbalance entre las clases es notable, se implementó el algoritmo W-SVM para proveer al clasificador SVM de una herramienta que le permita un aprendizaje basado en criterios de sensibilidad de costo extraídos de la poca información conocida que se tienen sobre la base de datos, así se podrá garantizar que haya representación de ambas clases en el entrenamiento de la máquina. Los parámetros a optimizar son: la constante de penalización (C), el peso de la clase 1 (W_1), y el peso de la clase 2 (W_2). Una vez obtenidos los mejores valores, se introducen en el clasificador SVM para su entrenamiento como se puede apreciar en la metodología propuesta.

El algoritmo de optimización por enjambre de partículas (PSO) fue implementado para sintonizar los parámetros de todos los algoritmos implementados en los diferentes bloques que componen la metodología. Los parámetros sintonizados fueron: σ para la función kernel gaussiana y para las métricas de distancia geométricas, $K_m y\, m$ para las métricas bio-inspiradas Mismatch y Gappy, y K_n fue sintonizado para el neighbohood kernel.

Para implementar el algoritmo PSO se necesita implementar una validación cruzada interna de diez particiones. Los datos de entrenamiento obtenidos de la matriz kernel semi-supervisada son divididos en dos particiones, 90% para los datos de entrenamiento de la validación cruzada interna y el otro 10% para los datos de prueba de la validación cruzada interna.

El algoritmo PSO inicializa algunos parámetros como candidatos para que los hiper-parámetros sean sintonizados. Luego con estos parámetros una SVM es entrenada y probada para cada partición. Cuando la validación cruzada interna de diez particiones se completa, se calcula una matriz de confusión.

Luego la media geométrica entre la sensibilidad y la especificidad es retornada hacia el algoritmo PSO. Este algoritmo PSO se repite hasta que se logra maximizar el valor de la media geométrica. Los parámetros que dieron los mejores resultados para el modelo de predicción de la SVM interna y dieron el valor máximo de la media geométrica, son retornados por el algoritmo PSO.

Los valores retornados por el PSO son los hiper-parámetros óptimos que servirán como parámetros de entrada para realizar el entrenamiento de la SVM (con los datos de entrada iniciales), después este modelo se utiliza para predecir la clase de los datos iniciales de prueba en cada iteración de la validación cruzada externa.

Cuando se finaliza con la validación cruzada externa de diez particiones, se calcula una matriz de confusión con los valores predichos por los modelos de la SVM. La media geométrica entre la sensibilidad y la especificidad se retorna como medida global de desempeño obtenido en el problema de clasificación.

Todos los algoritmos fueron realizados en el software libre de computación estadística, R. Para el clasificador SVM se utilizó el paquete “kernlab”, mientras que los kernels de secuencias se calcularon usando el paquete “kebabs” y para el PSO se utiliza el paquete “pso”. Todos los paquetes están disponibles de manera gratuita en el proyecto R-CRAN.

3.6 Metodología propuesta

3.6.1 Datos de entrada

Se comenzó con un conjunto de 3368 secuencias proteicas, cada una con 544 características, a este conjunto de datos se le aplicaron algoritmos de extracción de características, con lo que se construyó un espacio de representación con 3368 muestras, cada una con 438 características discriminantes para la clasificación, que contienen atributos organizados en tres grandes grupos: físico-químicos (obtenidos a partir del cómputo con la base de datos de proteínas “aa.index”), composición de la primera

estructura (Buxbaum, 2007) y composición de la segunda estructura de las proteínas (Frishman & Argos, 1997).

3.6.2 Matriz de distancias

Se construyó una matriz de distancias a la cual se le introdujo una métrica de distancia de origen geométrico o bioinspirado. Las métricas de distancia geométricas fueron la Euclídea y la Mahalanobis, y las medidas de similitud bioinspiradas fueron Mismatch y Gappy. De acuerdo a la métrica de distancia seleccionada, se tienen dos caminos para generar la matriz de distancias, así, en el caso de que sea geométrica, los datos deben tomarse después de haberse realizado la extracción y selección de características, por el contrario, si la métrica es bioinspirada, los datos se toman directamente del 90% de los datos seleccionados de la base de datos principal.

3.6.3 Algoritmo Neighborhood Kernel

Este algoritmo se utiliza para la construcción de un clasificador con kernel semi-supervisado, su función principal es cambiar la métrica de distancia de tal manera que la distancia relativa entre dos puntos es mucho más pequeña si los puntos están en el mismo clúster, lo cual se logra haciendo un cambio del espacio de representación del clasificador, teniendo en cuenta la estructura descrita por los datos no etiquetados. El Neighborhood kernel utiliza un promedio de más de un vecino de secuencias definidas por una medida de similitud. Lo que hace es calcular una medida de similitud de secuencias vecinas o neighborhood Nbd (x) para cada secuencia de entrada x, que representa el promedio de los vectores de características para los miembros de su vecindario o neighborhood.

3.6.4 Matriz Kernel semi-supervisada

La matriz kernel semi-supervisada se forma computando la matriz kernel con la matriz de distancias. La matriz kernel se construye a partir de la base de datos Embryophyta, de la cual se selecciona el 90% de los datos de manera aleatoria, a los que se les extrae y seleccionan las características más relevantes. Las 14 matrices kernel obtenidas tienen una reducción de sus dimensiones y una función de distancia de base radial (RBF). Por último se aplica el algoritmo Neighborhood Kernel entre la matrices construidas para que éste genere la matriz kernel semi-supervisada.

3.6.5 Algoritmo PSO

La sintonización de los hiper-parámetros se lleva a cabo con el algoritmo PSO (Kennedy & Eberhart, 1995), allí se desarrolla el algoritmo de optimización W-SVM (Yang, Song & Wang, 2007) para el problema de desbalance entre las clases basado en criterios de sensibilidad de costo, este algoritmo entrega los parámetros $\boldsymbol{C}$ **(**coeficiente de regularización**),** y los pesos $\boldsymbol{W_1}$ $\boldsymbol{y}$ $\boldsymbol{W_2}$ (pesos de las clases) para el clasificador SVM. Otros parámetros sintonizados en el bloque PSO fueron: σ para la función kernel gaussiana y para las métricas de distancia geométricas, $K_m y\, m$ para las métricas bio-inspiradas Mismatch y Gappy, y K_n fue sintonizado para el Neighbohood kernel. Una vez obtenidas las matrices kernel y de distancias, se introducen en el algoritmo Neighborhood Kernel para que éste genere la matriz kernel semi-supervisada.

3.6.6 Validación cruzada

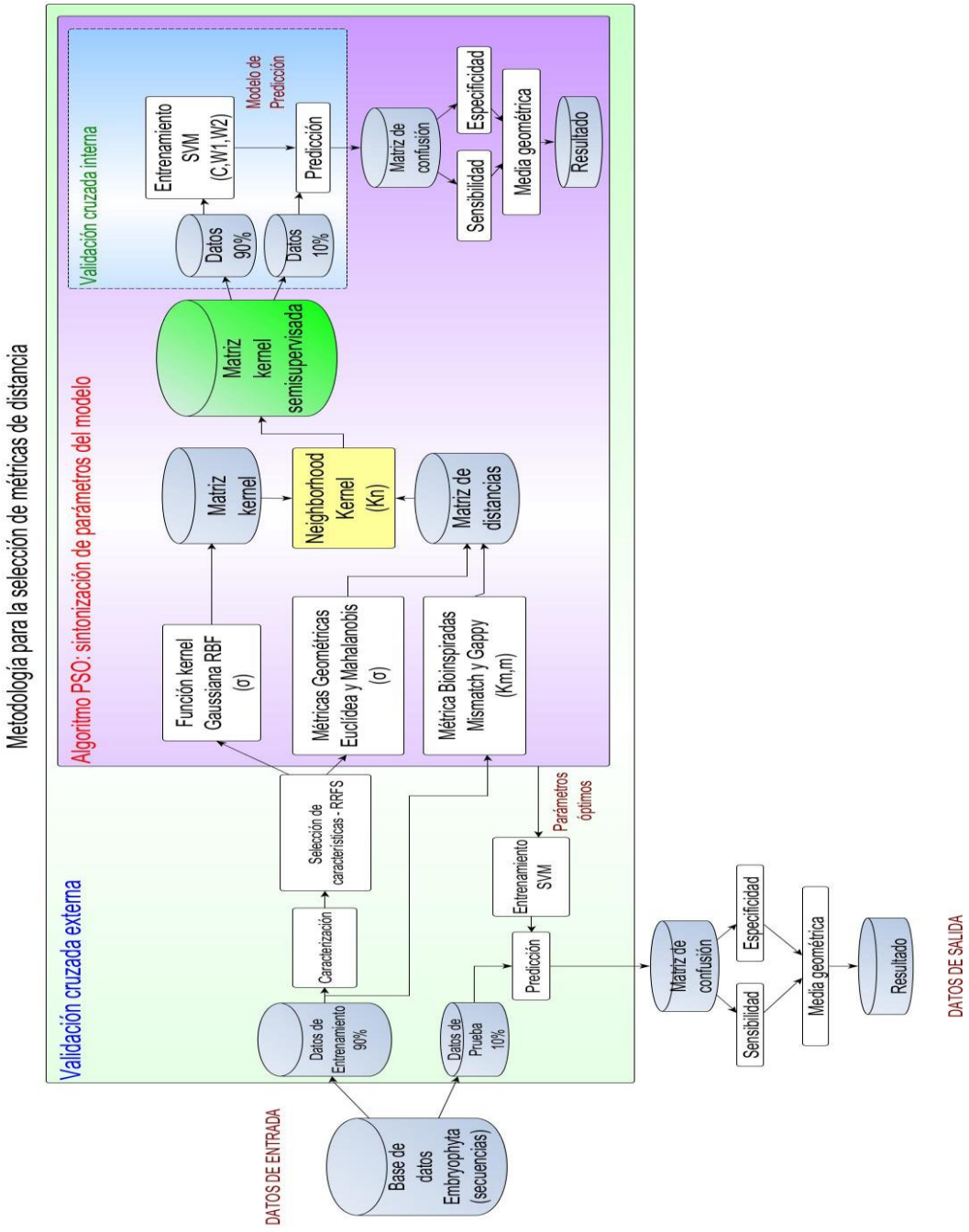
Para validar estadísticamente el modelo de predicción propuesto se usó la técnica de validación cruzada (Duan, K., et al., 2003) para garantizar una independencia entre los datos de entrenamiento y los datos de prueba. La validación cruzada interna se realizó en el algoritmo PSO con diez particiones, cuyos datos fueron obtenidos de la matriz kernel semi-supervisada. Una vez obtenidos los hiper-parámetros entregados por el algoritmo PSO, servirán como parámetros de entrada para realizar el entrenamiento de la SVM (con

los datos de entrada iniciales), después este modelo se utiliza para predecir la clase de los datos iniciales de prueba en cada iteración de la validación cruzada externa.

3.6.7 Evaluación del desempeño del clasificador

Una vez se finaliza la validación cruzada externa de diez particiones, se calcula una matriz de confusión (Chawla, N. V., et al, 2002) de la que se obtienen la sensibilidad y especificidad para calcular posteriormente la media geométrica con lo que evalua el desempeño de la SVM en los 14 problemas de clasificación.

Figura 10. Metodología para la Selección de la Métrica de Distancia

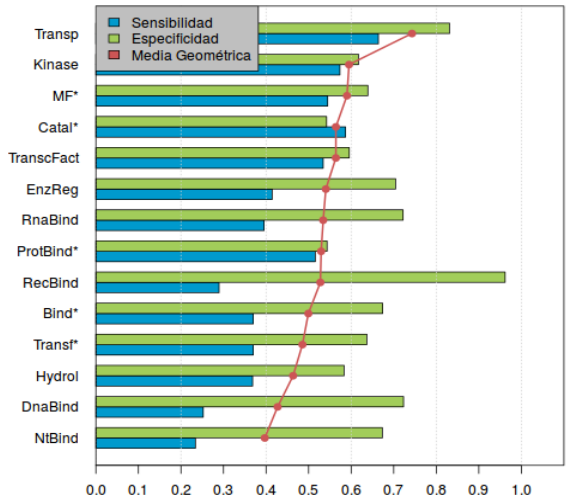


4. Resultados

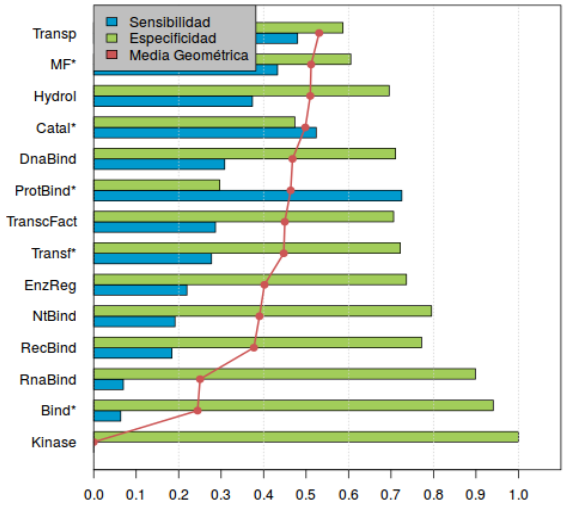
En la Figura 11 se presentan los resultados obtenidos de la clasificación, en ellos se grafican las 14 funciones moleculares o problemas de clasificación, la sensibilidad, la especificidad y la medida de desempeño basada en la media geométrica, organizadas de mayor a menor desempeño.

Figura 11. Medidas de distancia Geométricas y Bioinspiradas

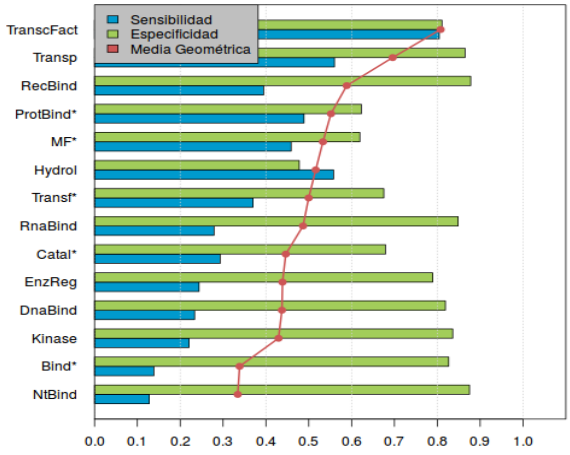
Medida de distancia Geométrica: Euclídea



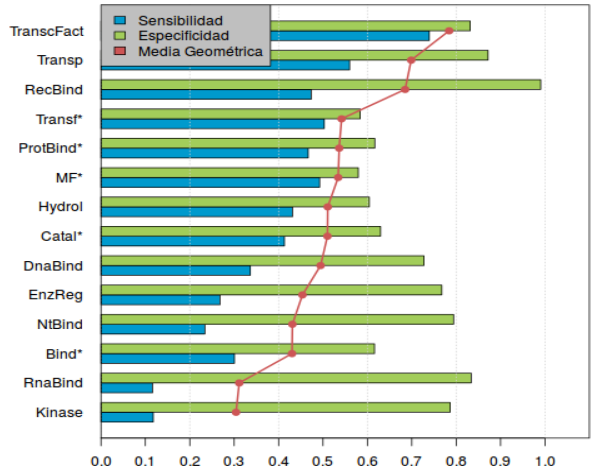
Medida de distancia Geométrica: Mahalanobis



Medida de distancia Bio-inspirada: Mismatch



Medida de distancia Bio-inspirada: Gappy



Aunque la función Transp tuvo los más altos porcentajes de acierto en todas las métricas, en la Euclídea se obtuvo un mejor desempeño, siendo este de 66.4%, lo que permite establecer que, en términos generales, cualquier métrica permite clasificarla acertadamente. Se observa que el peor desempeño del clasificador se da para la función Kinase, siendo sus porcentajes de acierto de 0, 42.9 y 30% para Mahalanobis, Mismatch y Gappy, respectivamente, sin embargo, la Función Euclídea permite clasificarla en el segundo lugar de mejores desempeños, con un 59.4%.

El peor desempeño fue el de la métrica Mahalanobis, donde las funciones NtBind, RnaBind, RecBind, Bind* y Kinase tuvieron un desempeño muy pobre, por debajo del 40%, llegando a ser nulo en la clasificación de Kinase.

Con las métricas bio-inspiradas se obtuvo un comportamiento muy similar, donde coinciden los tres mejores desempeños con las funciones TranscFact, Transp y RecBind, respectivamente. En Gappy se puede observar que mejoró la clasificación de funciones como DnaBind, NtBind, Bind*, sin embargo el acierto empeoró notablemente en RnaBind y Kinase.

Figura 12. Comparación de Medias Geométricas

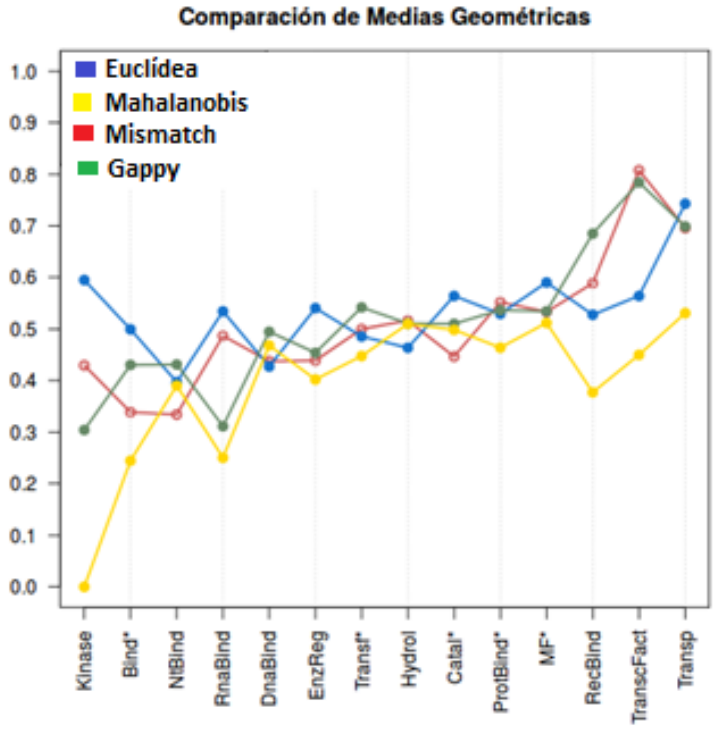


Figura 13. Comparación de Sensibilidades

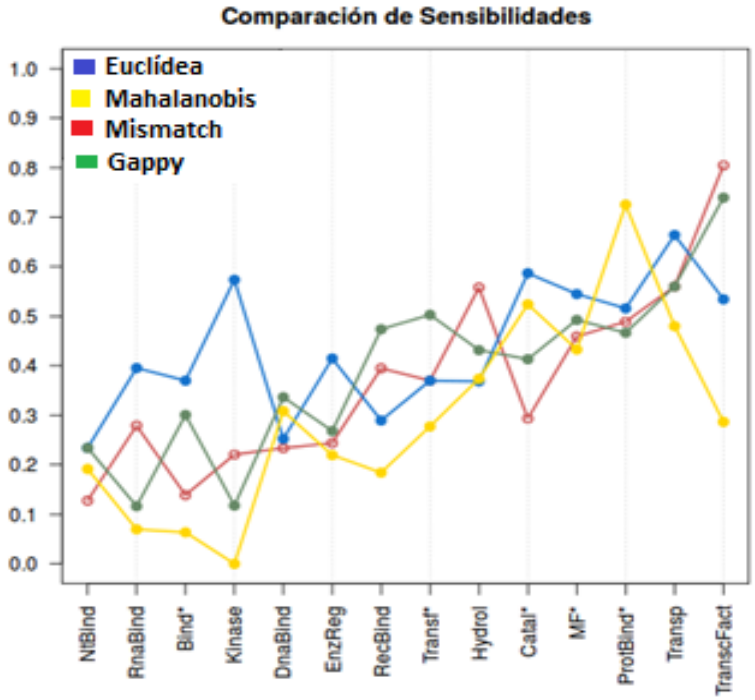
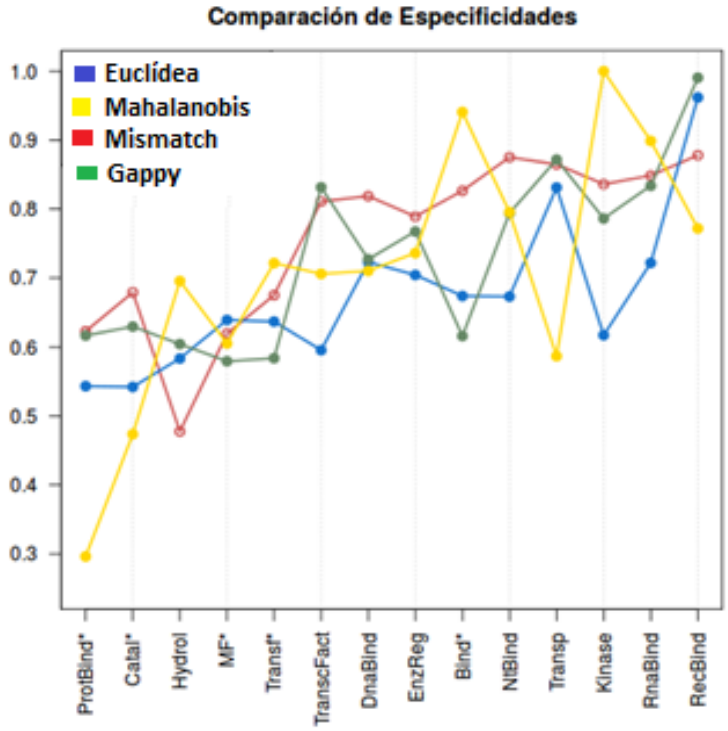


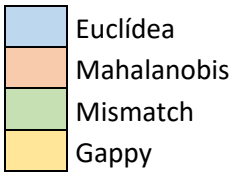
Figura 14. Comparación de Especificidades



En la Tabla 6 se tienen los desempeños del clasificador, organizados de mayor a menor y discriminados por funciones moleculares. Se utilizó una convención de colores para identificar cada métrica, así, el color azul se utilizó para identificar la métrica euclídea, el color rosado se utilizó para la Mahalanobis, el color verde para la Mismatch y el color naranja, para la Gappy. Se puede observar que la métrica Euclídea permitió los mejores desempeños para 7 de los 14 problemas, le sigue la Gappy, que permitió los mejores aciertos para 4 problemas, y por último, Mismatch, con el mejor desempeño en 3 funciones.

Tabla 7. Desempeño del clasificador por cada función molecular.

Función Molecular	Desempeño 1	Desempeño 2	Desempeño 3	Desempeño 4
NtBind	43.11%	39.70%	39.03%	33.43%
MF*	59.01%	53.42%	53.32%	51.18%
DnaBind	49.47%	46.81%	43.74%	42.72%
TranscFact	80.78%	78.42%	56.41%	44.99%
RnaBind	53.43%	48.66%	31.14%	25.04%
Catal*	56.41%	51.01%	49.82%	44.64%
RecBind	68.50%	58.87%	52.77%	37.71%
Transp	74.30%	69.87%	69.59%	53.07%
Bind*	49.94%	43.03%	33.86%	24.46%
ProtBind*	55.19%	53.65%	52.95%	46.38%
Kinase	59.50%	42.95%	30.42%	0.00%
Transf*	54.19%	49.98%	48.55%	44.75%
Hydrol	51.63%	51.07%	50.99%	46.36%
EnzReg	54.05%	45.38%	43.87%	40.20%



En la Tabla 7 se resume el desempeño del clasificador por funciones moleculares, discriminado de mayor a menor e identificando la métrica con la que se logró tal desempeño. El mejor porcentaje de acierto fue logrado para clasificar la función TranscFact con un 80.78% usando Mismatch, le sigue la función Transp con un 74.3% usando Euclídea, posteriormente se tiene RecBind con un 68.5% usando Gappy.

Tabla 8. Mejores desempeños del clasificador por funciones.

Función Molecular	Porcentaje de acierto	Métrica de distancia
TranscFact	80.78%	MISMATCH
Transp	74.30%	EUCLÍDEA
RecBind	68.50%	GAPPY
Kinase	59.50%	EUCLÍDEA
MF*	59.01%	EUCLÍDEA
Catal*	56.41%	EUCLÍDEA
ProtBind*	55.19%	MISMATCH
Transf*	54.19%	GAPPY
EnzReg	54.05%	EUCLÍDEA
RnaBind	53.43%	EUCLÍDEA
Hydrol	51.63%	MISMATCH
Bind*	49.94%	EUCLÍDEA
DnaBind	49.47%	GAPPY
NtBind	43.11%	GAPPY

Con excepción de la función DnaBind (49.47%), la métrica Euclídea permitió porcentajes de acierto por encima del 50% para 6 de los 14 problemas de clasificación. Por su parte, Mismatch también ofreció desempeños por encima del 50% para 3 problemas, lo que no ocurrió con Gappy, que, aunque aparece como primer lugar de rendimiento para 4 funciones, solo en dos de ellas (Transf* y RecBind) tiene un porcentaje de acierto mayor al 50%. Por último, Mahalanobis no representó un desempeño destacado en el clasificador para ninguno de los problemas. El mejor desempeño de la máquina usando esta métrica ni siquiera superó el 50% para clasificar la función DnaBind.

En la Tabla 8 se recomienda, de acuerdo a los resultados obtenidos, la métrica de distancia más apropiada para clasificar cada una de las funciones moleculares y según el desempeño, organizado de mayor a menor.

Tabla 9. Mejores desempeños del clasificador por métricas.

Métrica	Función	Desempeño
Euclídea	**Transp**	74.30%
	Kinase	59.50%
	MF*	59.01%
	Catal*	56.41%
	EnzReg	54.05%
	RnaBind	53.43%
	Bind*	49.94%
Gappy	**RecBind**	68.50%
	Transf*	54.19%
	DnaBind	49.47%
	NtBind	43.11%
Mismatch	**TranscFact**	80.78%
	ProtBind*	55.19%
	Hydrol	51.63%

Con el fin de tener una comparación con otros métodos encontrados en la literatura, se confrontan los resultados obtenidos con los resultados presentados en (García-López et al, 2013), quienes usaron la misma base de datos de organismos Embryophyta utilizada en el presente trabajo.

Allí se realiza la predicción de funciones mediante aprendizaje supervisado por medio de máquinas de vectores de soporte y criterios de sensibilidad de costo, el mismo algoritmo implementado en este trabajo para el tratamiento del problema de desbalance de clases, estando entonces en igualdad de condiciones para evaluar el desempeño entre un aprendizaje supervisado y uno semi-supervisado.

Tabla 10. Media Geométrica Neighborhood Vs. CS (García-López)

Función	Desempeño semi-supervisado (Neighborhood)	Desempeño supervisado (CS)
DnaBind	49.47%	51.9%
TranscFact	80.78%	62.9%
Catal*	56.41%	29.2%
Transp	74.30%	56.2%
Hydrol	51.63%	18.8%
EnzReg	54.05%	61.3%
Transf*	54.19%	49.3%

En (García-López et al, 2013) utilizaron un conjunto de datos con 1098 secuencias extraídas del Dataset Embryophyta, cada una de ellas con 438 características o atributos físico-químicos para predecir siete (7) funciones biológicas. En la Tabla 9 se puede observar cómo el desempeño con aprendizaje semi-supervisado superó notablemente al supervisado en cinco de las siete funciones, con porcentajes de acierto mayores en un 17.88%, 27.21%, 18.1%, 32.83% y 4.89%, en las funciones TranscFact, Catal, Transp, Hydrol y Transf*, respectivamente. Por otro lado, las funciones en las que el clasificador CS superó al clasificador semi-supervisado Neighborhood, tuvieron desempeños superiores en un 2.43% y 7.25%, en las funciones DnaBind y EnzReg, respectivamente.

Se puede concluir, que si bien la base de datos es la misma, y se utiliza el mismo algoritmo por sensibilidad de costo para balancear las clases, la inclusión de muestras sin etiquetar en el entrenamiento del clasificador es aprovechado para la construcción de un mejor modelo predictivo gracias a la información biológica implícita que hay en ellas.

5. Conclusiones y recomendaciones

5.1 Conclusiones

- La variabilidad en los resultados de cada clase es debido a que cada una de ellas origina un problema de clasificación diferente con particularidades especiales. Esto se evidencia desde el análisis de la cantidad de datos por clase (que se puede observar en la tabla 4), que además implica un porcentaje de desbalance asociado a cada problema. Mientras la medida de distancia Euclídea representa una comparación de los datos desde un punto de vista geométrico (distancias entre las muestras en el espacio de características), lo cual resultó ser eficiente para la mayoría de clases, las distancias Mismatch y Gappy representan una distancia más orientada a la interpretación biológica de los datos (alineamiento de secuencias), lo cual da cuenta de la distancia evolutiva entre los miembros de las categorías. Resulta importante resaltar que tres de las clases con mayor radio de desbalance (anclaje de nucleótidos, anclaje de ADN y anclaje de receptores) se vieron más beneficiadas por esta distancia con orientación biológica que por la distancia euclídea. Esto puede estar asociado con el hecho de que estas clases, por tener menor cantidad de muestras, son menos heterogéneas en su estructura primaria.

- Se comenzó con un conjunto de 3368 secuencias proteicas, cada una con 544 características, a este conjunto de datos se le aplicaron algoritmos de extracción de características, con lo que se construyó un espacio de representación con 3368 muestras, cada una con 438 características discriminantes para la clasificación, que contienen atributos organizados en tres grandes grupos: físico-químicos (obtenidos a partir del cómputo con la base de datos de proteínas “aa.index”), composición de la primera estructura (Buxbaum, 2007) y composición de la segunda estructura de las proteínas (Frishman & Argos, 1997).

- Se construyó una matriz de distancias a la cual se le introdujo una métrica de distancia de origen geométrico o bioinspirado. Las métricas de distancia geométricas fueron la

Euclídea y la Mahalanobis, y las medidas de similitud bioinspiradas fueron Mismatch y Gappy.

- De acuerdo a la métrica de distancia seleccionada, se tienen dos caminos para generar la matriz de distancias, así, en el caso de que sea geométrica, los datos deben tomarse después de haberse realizado la extracción y selección de características, por el contrario, si la métrica es bioinspirada, los datos se toman directamente del 90% de los datos seleccionados de la base de datos principal.

- La sintonización de los hiper-parámetros se logró mediante los algoritmos W-SVM (Yang, Song & Wang, 2007) y PSO (Kennedy & Eberhart, 1995). El primero se utilizó para el problema de desbalance entre las clases, con este se logró un aprendizaje basado en criterios de sensibilidad de costo, y el segundo, permitió sintonizar todos los parámetros de los algoritmos implementados en los diferentes bloques que componen la metodología. Los parámetros sintonizados fueron: σ para la función kernel gaussiana y para las métricas de distancia geométricas, $K_m, y\ m$ para las métricas bio-inspiradas Mismatch y Gappy, y K_n fue sintonizado para el Neighbohood kernel. Una vez obtenidas las matrices kernel y de distancias, se introducen en el algoritmo Neighborhood Kernel para que éste genere la matriz kernel semi-supervisada.

- Para validar estadísticamente el modelo de predicción propuesto se usó la técnica de validación cruzada (Duan, K., et al., 2003), con esta se garantiza una independencia entre los datos de entrenamiento y los datos de prueba. Se implementó una validación cruzada interna de diez particiones dentro del algoritmo PSO, cuyos datos fueron obtenidos de la matriz kernel semi-supervisada. Los valores retornados por el PSO son los hiper-parámetros óptimos que servirán como parámetros de entrada para realizar el entrenamiento de la SVM (con los datos de entrada iniciales), después este modelo se utiliza para predecir la clase de los datos iniciales de prueba en cada iteración de la validación cruzada externa.

- Cuando se finaliza con la validación cruzada externa de diez particiones, se calcula una matriz de confusión con los valores predichos por los modelos de la SVM. La media geométrica entre la sensibilidad y la especificidad se retorna como medida global de desempeño obtenido en el problema de clasificación.

- Para estimar la medida de desempeño de la SVM en los 14 problemas de clasificación, se calculó una matriz de confusión (Chawla, N. V., et al, 2002), de donde se obtiene la sensibilidad y la especificidad, con estas se calcula la media geométrica, la cual representa el balance entre las muestras bien clasificadas de la clase positiva y las muestras bien clasificadas de la clase negativa (Tang, Y., et al., 2010).

- Se comprobó que la selección apropiada de la métrica de distancia es vital para mejorar el rendimiento de clasificadores en la notación de las diferentes funciones moleculares. Esto representa un importante aporte al desarrollo de nuevas y mejores herramientas de análisis a nivel proteómico, usando el aprendizaje de máquina para predecir con mayor acierto organismos vegetales, puesto que en la actualidad presentan muy bajo rendimiento en comparación con otras herramientas diseñadas sobre organismos modelo, debido en parte a la escasa cantidad de muestras etiquetadas que resultan insuficientes para entrenar algoritmos supervisados que tengan alta capacidad de generalización.

- A pesar del costo computacional de este tipo de métodos, por razones como la alta dimensionalidad de los datos, por el gran número de muestras no etiquetadas, por el alto desbalance que hay entre las clases y porque una misma secuencia puede pertenecer simultáneamente a varias funciones biológicas, su nivel de acierto es muy satisfactorio comparado con los métodos tradicionales de alineamiento de secuencias.

- Este tipo de soluciones supera a métodos de notación biológica tradicionales como BLAST y PSI-BLAST, porque estos son limitados para realizar búsquedas masivas de similitudes entre secuencias por ser un recurso compartido, porque no permiten personalizar las bases de datos, ni aplicarle etapas de pre-procesamiento o de optimización para mejorar el desempeño. Para el caso puntual de la clasificación del conjunto extraído de Embryophyta, cabe anotar que debido a las altas dimensiones del número de muestras, como del número de características, el uso de estos algoritmos no es el más indicado.

- La complejidad para comparar secuencias de proteínas está en que se valoran positivamente también los parecidos de aminoácidos que no son idénticos, pero que son químicamente similares, lo que conduce a que los alineamientos de las secuencias

al azar puedan confundirse con alineamientos realmente significativos. Por otro lado, cabe anotar que otra de las complejidades en la clasificación de secuencias de proteínas radica en el hecho de que una misma secuencia puede pertenecer simultáneamente a varias funciones biológicas.

- Los enfoques entre un aprendizaje supervisado y uno semi-supervisado son diferentes. Aunque en ambos se construye una función matemática para la clasificación, con el primero se entrena el clasificador con un conjunto de datos cuya clase es conocida, mientras que con el segundo se construye el conjunto de entrenamiento con datos etiquetados y no etiquetados, siendo estos últimos de mayor volumen para que ayuden a la construcción de un mejor modelo predictivo. Este trabajo de aprendizaje semi-supervisado se presenta como una herramienta computacional que permita automatizar el proceso de clasificación del alto volumen de datos biológicos sin etiquetar, existentes en la naturaleza, en este caso proteínas de plantas terrestres.

5.2 Recomendaciones y Trabajo futuro

- Usar nuevos métodos de balanceo de clases.
- Usar otras técnicas de clasificación semi-supervisadas.
- Implementar otras métricas de distancia para comparar secuencias de proteínas y comparar los resultados con métodos de alineamiento de secuencias tradicionales.
- Ampliar la base de datos y probar con otras bases de datos biológicas.

Bibliografía

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space (pp. 420-434). Springer Berlin Heidelberg.

Arango-Argoty, G. A., Jaramillo-Garzón, J. A., Röthlisberger, S, Castellanos-Domínguez, C. G. (2011). Classification of unaligned sequences based on prototype motifs representation. Proceedings of the 6th Colombian Computing Congress (6CCC). ISBN: 978-1-4577-0286-0.

Arango-Argoty, G., Giraldo-Forero, A. F., Jaramillo-Garzón, J. A., Duque-Muñoz, L., & Castellanos-Domínguez, G. (2013). Predicting Molecular Functions in Plants using Wavelet-based Motifs. In BIOINFORMATICS (pp. 140-145).

Al-Shahib, A., Breitling, R., & Gilbert, D. R. (2007). Predicting protein function by machine learning on amino acid sequences--a critical evaluation. BMC genomics, 8, 78. doi:10.1186/1471-2164-8-78

Audesirk, T., Audesirk, G., & Byers, B. E. (2003). Biología: La vida en la Tierra. Pearson educación.

Avalos García, A., & Pérez-Urria Carril, E. (2011). Metabolismo secundario de plantas. Reduca (Biología), 2(3).

Ayyash, M. (2012). A framework for a Minkowski distance based multi metric quality of service monitoring infrastructure for mobile ad hoc networks. International Journal on Electrical Engineering and Informatics, 4(2), 289-305.

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple Kernel Learning. Conic Duality , and the SMO Algorithm. Electrical Engineering.

Baghshah, M. S. (2010). Low-Rank Kernel Learning for Semi-supervised Clustering. Electrical Engineering.

Baldi, P. y Brunak, S. (2001). Bioinformatics: The Machine Learning Approach. MIT Press.

Ball, G. R., & Srihari, S. N. (2009). Semi-supervised Learning for Handwriting Recognition. 2009 10th International Conference on Document Analysis and Recognition, 26-30.

Barreto, E. (2008). Bioinformática: una oportunidad y un desafío. Revista Colombiana de Biotecnología, 10(1), 132-138.

Bi, R., Zhou, Y., Lu, F., & Wang, W. (2006). Predicting Gene Ontology functions based on support vector machines and statistical significance estimation. Neurocomputing, 70, 718-725. doi:10.1016/j.neucom. 2006.10.006

Bieto, J. A., Cubillo, M. T., Mangas, I. B., & Ormaechea, A. G. (2008). Fundamentos de fisiología vegetal. McGraw-Hill Interamericana de España.

Bodo, Z. (2008). Hierarchical cluster kernels for supervised and semi-supervised learning. 2008 4th International Conference on Intelligent Computer Communication and Processing, 9-16.

Buxbaum, E. Fundamentals of protein structure and function. Springer. 2007

Cai, C. Z. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Research, 31(13), 3692-3697. doi:10.1093/nar/gkg600

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., et al. (2004). The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. Nucleic acids research, 32(Database issue), D262--6. doi:10.1093/nar/gkh021

Cerón, A., Leal, M. & Nassar, F. (2008). ¿Hay futuro para la economía colombiana en la biodiversidad?. Revista EAN No.62 enero-abril de 2008 p.107-124.

Chapelle, O., & Zien, A. (2005). Semi-Supervised Classification by Low Density Separation. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (pp. 57–64).

Chapelle, O., Chi, M., & Zien, A. (2006). A continuation method for semi-supervised SVMs. *Proceedings of the Twenty-Third International Conference on Machine Learning.*

Chapelle, O., Schölkopf, B. & Zien, A. (2006). *Semi-Supervised Learning*. London, U.K.: MIT Press. Chapelle, O., Schölkopf, B. & Zien, A. editors.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16(1), 321-357.

Chen, K., & Wang, S. (2011). Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. IEEE transactions on pattern analysis and machine intelligence, 33(1), 129-43.

Cheng, V., & Li, C. H. (2006). Personalized Spam Filtering with Semi-supervised Classifier Ensemble Classification of emails using SVM and naïve Bayes Classifier. Intelligence, 0-6.

Chou, K.-C., & Shen, H.-B. (2010). Plant-mPLoc: a top-down strategy to augment the power for predicting plant protein subcellular localization. PloS one, 5(6), e11335. doi:10.1371/journal.pone.0011335

Conesa, A., & Götz, S. (2008). Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics. International journal of plant genomics, 2008, 619832. doi:10.1155/2008/619832

Conn, E., Stumpf, P. K., Bruening, G., Doi, R. (2005). Bioquímica fundamental.

Cuadras, C. (1989). Distancias Estadísticas. Estadística Española. Departament d'Estadística. Universitat de Barcelona 30(119):295-378

Dai, G., & Yeung, D.-Y. (2007). Kernel selection for semi-supervised kernel machines. Proceedings of the 24th international conference on Machine learning - ICML '07, 185-192. New York, New York, USA: ACM Press.

Damodaran, S. (2000). Aminoácidos, péptidos y proteínas. Química de alimentos, 2, 490.

Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing, 51, 41-59.

Eberhart, R. C., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science (Vol. 1, pp. 39-43).

Eberhart, R. C., & Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 81-86). IEEE.

Emanuelsson, O. and Nielsen, H. and Brunak, S. and von Heijne, G. (2000). Predicting subcellular localization of proteins based on their N-terminal amino acid sequence, Journal of molecular Biology 300 (4). Elsevier.

Eskin, E., Weston, J., Noble, W. S., & Leslie, C. S. (2002). Mismatch string kernels for SVM protein classification. In Advances in neural information processing systems (pp. 1417-1424).

Frishman, D. and Argos, P. (1997). Seventy-five percent accuracy in protein secondary structure prediction. Proteins Struct Funct and Genet. 27(3):329–335.

García-López, S., Jaramillo-Garzón, J. A., & Castellanos-Dominguez, G. (2013). Optimization of Cost Sensitive Models to Improve Prediction of Molecular Functions. In Biomedical Engineering Systems and Technologies (pp. 207-222). Springer Berlin Heidelberg.

Giraldo-Forero, A. Jaramillo-Garzón, J. A., Röthlisberger, S, Castellanos-Domínguez, C. G. (2011). Análisis de la capacidad de generalización a inter-especies en la predicción de ubicaciones o subcelulares de proteínas. Memorias del XVI Simposio de Tratamiento de Señales, Imágenes y Visión Artificial, Cali, Colombia, 2011. ISBN: 978-958-8347-55-4.

Giraldo-Forero, A. F., Jaramillo-Garzón, J. A., Ruiz-Muñoz, J. F., & Castellanos-Domínguez, C. G. (2013). Managing imbalanced data sets in multi-label problems: a case study with the SMOTE algorithm. In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (pp. 334-342). Springer Berlin Heidelberg.

Groth, D., Lehrach, H., & Hennig, S. (2004). GOblet: a platform for Gene Ontology annotation of anonymous sequence data. Nucleic acids research, 32(Web Server issue), W313--7. doi:10.1093/nar/gkh406.

Hannagan, T., & Grainger, J. (2012). Protein Analysis Meets Visual Word Recognition: A Casefor String Kernels in the Brain. Cognitive Science Laboratory, CNRS Aix-Marseille University.

Handl, J., & Knowles, J. (2005). Multiobjective clustering around medoids Dev (C) = E E 6 (i , Pk),. Computational Complexity, 632-639.

Hawkins, T., Chitale, M., Luban, S., & Kihara, D. (2009). PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. Proteins, 74(3), 566-582. doi:10.1002/prot.22172

Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences, 89(22), 10915-10919.

Hernández, N. (2013). Métodos de Kernels en secuencias para la clasificación de residuos catalíticos en sitios activos de enzimas (Doctoral dissertation, Universidad Nacional de Colombia).

Holman, R. M., Robins, W. W. (1961). Botánica general (No. QK 47. H6418).

Jain, E., Bairoch, A., Duvaud, S., Phan, I., Redaschi, N., Suzek, B. E., Martin, M. J., et al. (2009). Infrastructure for the life sciences: design and implementation of the UniProt website. BMC bioinformatics, 10, 136. doi:10.1186/1471-2105-10-136

Jaramillo-Garzón, J. A.. Arango-Argoty, G. A. , Röthlisberger, S, Castellanos-Domínguez, C. G. (2011). Prediction of protein subcellular localization based on variable-length motifs detection and dissimilarity based classification. In Proceedings of the 33nd Annual International Conference of the IEEE EMBS, Boston, Estados Unidos. ISSN: 1557-170X.

Jaramillo-Garzón, J., Gallardo-Chacón, J., Castellanos-Domínguez, C. and Perera-Lluna, A. (2013). Predictability of gene ontology slim-terms from primary structure information in *Embryophyta* plant proteins. BMC Bioinformatics 2013. 14:68.

Jensen, L. J. (2003). Prediction of human protein function according to Gene Ontology categories. Bioinformatics, 19(5), 635-642. doi:10.1093/bioinformatics/btg036

Joachims, T. (1999). Transductive inference for text clas- sification using support vector machines. *Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 41–48).

Jones, C. E., Schwerdt, J., Bretag, T. A., Baumann, U., & Brown, A. L. (2008). GOSLING: a rule-based protein annotator using BLAST and GO. Bioinformatics (Oxford, England), 24(22), 2628-2629. doi:10.1093/bioinformatics/btn486

Jung, J., & Thon, M. R. (2008). Gene function prediction using protein domain probability and hierarchical Gene Ontology information. 2008 19th International Conference on Pattern Recognition, 1-4. Ieee. doi:10.1109/ICPR.2008.4761737

Jung, J., Yi, G., Sukno, S. a, & Thon, M. R. (2010). PoGO: Prediction of Gene Ontology terms for fungal proteins. BMC bioinformatics, 11, 215. doi:10.1186/1471-2105-11-215

Kennedy, J & Eberhart, C. (1995). “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., pp. 1942–1948

Khan, S. (2003). GoFigure: Automated Gene OntologyTM annotation. Bioinformatics, 19(18), 2484-2485. doi:10.1093/bioinformatics/btg338

King, R. D., Wise, P. H., & Clare, A. (2004). Confirmation of data mining based predictions of protein function. Bioinformatics (Oxford, England), 20(7), 1110-1118. doi:10.1093/bioinformatics/bth047.

King, B. R. (2009). Protein sequence classification with Bayesian supervised and semi-supervised learned classifiers. Evaluation. STATE UNIVERSITY OF NEW YORK AT ALBANY. Retrieved from http://gradworks.umi.com/33/38/3338851.html

Kokare, M., Chatterji, B. N., & Biswas, P. K. (2003, October). Comparison of similarity metrics for texture image retrieval. In TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region (Vol. 2, pp. 571-575). IEEE.

Kuksa, P., Huang, P. H., & Pavlovic, V. (2008). A fast, large-scale learning method for protein sequence classification. In 8th Int. Workshop on Data Mining in Bioinformatics (pp. 29-37).

Kuksa, P. P., Huang, P. H., & Pavlovic, V. (2009). Scalable algorithms for string kernels with inexact matching. In Advances in Neural Information Processing Systems (pp. 881-888).

Lance, G. N., & Williams, W. T. (1967). A general theory of classificatory sorting strategies 1. Hierarchical systems. The computer journal, 9(4), 373-380.

Leslie, C., Eskin, E., Cohen, A., Weston, J., & Noble, W. S. (n.d.). Mismatch string kernels for discriminative protein classification. Genome.

Leslie, C. S., Eskin, E., & Noble, W. S. (2002, January). The spectrum kernel: A string kernel for SVM protein classification. In Pacific symposium on biocomputing (Vol. 7, pp. 566-575).

Li, T., Yang, J., Peng, X., Chen, Z., & Luo, C. (2008). Prediction and Early Warning Method for Flea Beetle Based on Semi-supervised Learning Algorithm. 2008 Fourth International Conference on Natural Computation, 217-221.

Li, C.-hung, & Wu, Z.-li. (2004). Spectral Energy Minimization for Semi-supervised Learning, 13-21.

Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, 22(13):1658–1659, 2006.

Longstaff, B., Reddy, S., & Estrin, D. (2010). Improving Activity Classification for Health Applications on Mobile Devices using Active and Semi-Supervised Learning.

Magallóna, S., & Hilub, K. W. (2009). Land plants (embryophyta). The timetree of life, 133.

Manresa, R. E. (1983). La fertilización foliar con aminoácidos. Horticultura: Revista de industria, distribución y socioeconomía hortícola: frutas, hortalizas, flores, plantas, árboles ornamentales y viveros, (12), 33-35.

Martin, D. M. a, Berriman, M., & Barton, G. J. (2004). GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. BMC bioinformatics, 5, 178. doi:10.1186/1471-2105-5-178.

Mathews, C. K., Van Holde, K. E., & Ahern, K. G. (2002). Bioquímica. 3ra. Edición. Madrid: Adison Wesley Pearson Education.

Mei, S., & Fei, W. (2010). Amino acid classification based spectrum kernel fusion for protein subnuclear localization. BMC bioinformatics, 11(1), 1.

Mendialdua, A. Z. (2008). Aproximaciones a SVM semisupervisado multiclase para clasificación de páginas web Arkaitz Zubiaga Mendialdua.

Mika, S., Schölkopf, B., Smola, A., M, K.-robert, Scholz, M., & R, G. (n.d.). Kernel PCA and De-Noising in Feature Spaces. Analysis, (i).

Ministerio del Media Ambiente, Dirección Nacional de Planeación e Instituto Alexander von Humboldt. (1995). Política Nacional de Biodiversidad. Bogotá.

Ministerio de Salud. (1999). Resolucion número 1995 de 1999 "por la cual se establecen normas para el manejo de la historia clínica"

Muller, H., Michoux, N., Bandon, D. & Geissbuhler, A. (2004) A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. *International journal of medical informatics*.

Murray, R. K. (2010). Harper's illustrated biochemistry. Harper. Bioquímica ilustrada (No. QP514. M8718 2010).

Nabors, M. W., & González-Barreda, P. (2004). Introduction to botany. New York: Pearson Benjamin Cummings.

Olivares-Quiroz, L., & García-Colín, L. S. (2004). Plegamiento de las proteínas: Un problema interdisciplinario. Rev. Soc. Quím Méx, 48, 95-105.

Pandey, G., Kumar, V and Steinbach, M, (2006). Computational Approaches for Protein Function Prediction: A Survey. Twin Cities: Department of Computer Science and Engineering, University of Minnesota.

Pardo Arquero, V. P. (2004). La importancia de las vitaminas en la nutrición de personas que realizan actividad físicodeportiva. The importance of vitamins in the feeding of people who do physical and sportive activity.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods—support vector learning, 3.

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., ... & Pandey, G. (2013). A large-scale evaluation of computational protein function prediction. Nature methods, 10(3), 221-227.

Rhee, S.,Wood,V., Dolinski, K. & Draghici, S. Use and misuse of the gene ontology annotations. Nature reviews. Genetics, 9(7):509–515, 2008. ISSN 1471-0064. doi: 10.1038/nrg2363.

Rhee, S. Y., Beavis, W., Berardini, T. Z., Chen, G., Dixon, D., Doyle, A., ... & Miller, N. (2003). The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucleic acids research, 31(1), 224-228.

Se, U. S. (n.d.). Thanh Ngo and Rui Kuang Department of Computer Science and Engineering , University of Minnesota , Twin Cities ,. Science, 7-10.

Shimizu, K., Muraoka, Y., Hirose, S., Tomii, K., & Noguchi, T. (2007). Predicting mostly disordered proteins by using structure-unknown protein data. BMC bioinformatics, 8, 78.

Sindhwani, V., Keerthi, S., & Chapelle, O. (2006). Deter- ministic annealing for semi-supervised kernel machines. *Proceedings of the Twenty-Third International Conference on Machine Learning.*

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Proceedings of the Twenty-Second Interna- tional Conference on Machine Learning*.

Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., ... & Altschul, S. F. (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic acids research, 29(14), 2994-3005.

Schölkopf, B. & Smola, A. (2002). Learning with kernel. MIT Press.

Schölkopf, B. (2003). Kernel methods and Support Vector Machines. Max-Planck-Institut für biologische Kybernetik.

Schölkopf, B. & Smola, A. (2003). A short introduction to learning with kernels. Proceedings. LNAI, pp:41-64.

Small, I. and Peeters, N. and Legeai, F. and Lurin, C (2004). Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences. Proteomics, 4 (6), 1581-1590, Wiley Online Library.

Tang, Y., Zhang, Y. Q., Chawla, N. V., & Krasser, S. (2009). SVMs modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(1), 281-288.

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010, July). Cost-sensitive learning methods for imbalanced data. In Neural Networks (IJCNN), The 2010 International Joint Conference on (pp. 1-8). IEEE.

The Gene Ontology Consortium. The gene ontology (go) database and informatics resource. Nucleic Acids Res, 32:258–261, 2004.

Vadivel, A. K. M. S. S. A., Majumdar, A. K., & Sural, S. (2003, January). Performance comparison of distance metrics in content-based image retrieval applications. In International Conference on Information Technology (CIT), Bhubaneswar, India (pp. 159-164).

Vapnik, V. (1998) Statistical Learning Theory. Wiley-interscience.

Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., & Stafford, W. (2004). Semi-supervised protein classification using cluster kernels. Bioinformatics, 00(00), 1-8.

Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., & Noble, W. S. (2005). Semi-supervised protein classification using cluster kernels. Bioinformatics (Oxford, England), 21(15), 3241-7. doi: 10.1093/bioinformatics/bti497.

Wu, J., Diao, Y.-B., Li, M.-L., Fang, Y.-P., & Ma, D.-C. (2009). A semi-supervised learning based method: Laplacian support vector machine used in diabetes disease diagnosis. Interdisciplinary sciences, computational life sciences, 1(2), 151-5.

Yang, X., Song, Q., & Wang, Y. (2007). A weighted support vector machine for data classification. International Journal of Pattern Recognition and Artificial Intelligence, 21(05), 961-976.

Yao, J., Zhang, Z., Antani, S., Long, L. R. & Thoma, G. (2006). Automatic medical image annotation and retrieval using semi-secc. *Multimedia and Expo, IEEE International Conference on*, 0:2005-2008, 2006.

Yu, L., Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224.

Zehetner, G. (2003). OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. Nucleic Acids Research, 31(13), 3799-3803. doi:10.1093/nar/gkg555

Zhao, B., & Kwok, J. T. (n.d.). Multiple Kernel Clustering. Science, (Mmc).

Zhao, Z., & Wang, G. (2010). Semi-supervised Kernel Based Progressive SVM. 2010 Fourth International Conference on Genetic and Evolutionary Computing, (1), 102-105.

Zhao X, Chen L, Aihara K: Protein function prediction with high-throughput data. Amino acids 2008, 35:517-530.

Zhou, Z. (2006). *Learning with unlabeled data and its application to image retrieval*. *9th Pacific Rim International Conference on Artificial Intelligence*, (pp.5–10), Guilin, China.

Zhou, Z.-H., & Li, M. (2009). Semi-supervised learning by disagreement. Knowledge and Information Systems, 24(3), 415-439.

Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1-130.

Zubiaga, A., & Mart, R. (1997). Comparativa de Aproximaciones a SVM Semisupervisado Multiclase para Clasificación de Páginas Web.