Universidad Central "Marta Abreu" de las Villas

Facultad de Matemática-Física-Computación

Lic. Ciencias de la Computación

Trabajo de Diploma

Título: Ambiente para la búsqueda de combinaciones de modelos bases en multiclasificadores de Quimio-Bioinformática.

Autoras:

Maricel Meneses Gómez

Dámaris Cedré Gutiérrez

Tutores:

Dr.: Alfredo Meneses Marcel

Dr.: Yovany Marrero Ponce

Dra.: Gladys Casas Cardoso

Santa Clara, Cuba

2012-2013

DICTAMEN

Maricel Meneses Gómez y Dámaris Cedré Gutiérrez, hacemos constar que el trabajo titulado “Ambiente para la búsqueda de combinaciones óptimas de modelos bases en multiclasificadores de Quimio-Bioinformática” fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Licenciatura en Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor 1 Firma del autor 2

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor Firma del jefe del Laboratorio

Fecha

PENSAMIENTO

Saber no es suficiente, debemos aplicar.

Desear no es suficiente, debemos hacer.

Johann W. Von Goethe

DEDICATORIA

A mis padres por haberme dado la vida.

A mi hermana Susana por soportarme y quererme.

A mis abuelos por brindarme todo su amor y cariño.

A mi novio Alejandro por apoyarme en cada momento.

Maricel Meneses Gómez

A Dios por su presencia en mi vida.

A mis padres por su apoyo y dedicación.

A mi esposo por su amor y por confiar en mí.

A mis hermanos por su amor y cariño.

A mi abuela, Delfina.

Dámaris Cedré Gutiérrez

AGRADECIMIENTOS

A mis padres, por su amor infinito y el sacrificio realizado para mi formación profesional.

A mi hermana Susana, por ser mi punto de apoyo.

A mis abuelos, por todos los momentos que me han sacado adelante.

A mis tíos y primos que han dejado en mí recuerdos inolvidables.

A mi novio por su paciencia y por formar parte de mi vida.

A toda mi familia por creer en mí y por su ayuda excepcional.

Maricel Meneses Gómez

A Dios por darme las fuerzas cada día para llegar a la meta.

A mis padres por su amor incondicional y su sacrificio para formarme profesionalmente.

A mi esposo por su amor, por estar a mi lado y apoyarme cada instante.

A mis hermanos por su cariño, por su paciencia y por su confianza.

A mi abuela Delfina por tenerme presente en sus oraciones.

A toda mi familia por su preocupación, su cariño y apoyo incondicional.

Dámaris Cedré Gutiérrez

A nuestros tutores Gladys, Alfredo y Yovany por el tiempo dedicado y por sus valiosas ideas.

A Adis, Pino y Leydi que nos han bridado su ayuda excepcional para la realización de este trabajo.

A nuestros compañeros Elizabeth, Jorge Luis y Alejandro por su colaboración.

A todos aquellos que de una forma u otra han contribuido en el proyecto, gracias.

RESUMEN

RESUMEN

La búsqueda de medicamentos contra enfermedades que azotan a la salud humana sigue siendo una de las prioridades a nivel mundial. Dada la poca efectividad de los ensayos de "*prueba y error*", se hace necesaria la búsqueda de soluciones para el desarrollo de vías alternativas para el descubrimiento de nuevos compuestos activos. El empleo de multiclasificadores, (clasificadores que se fusionan y se obtiene como resultado la combinación de cada una de ellas) ha sido una estrategia muy efectiva que ha permitido una mayor precisión de los análisis y a la vez, la producción de buenos resultados en la búsqueda de toma de decisiones. En el presente trabajo se pretende profundizar en los métodos de ensamble o sistema de múltiples clasificadores, implementando un sistema que combina varios modelos de clasificación para obtener mejores predicciones. En la combinación de las salidas de los clasificadores de base se utilizan varias funciones matemáticas. La selección de los clasificadores de base se hace de acuerdo a un grupo de estadísticas que permiten comparar la exactitud de la predicción y un grupo de medidas de diversidad que combinan los resultados de varios clasificadores individuales y proponen combinaciones de los mismos. Se presentan los aspectos principales del análisis, diseño e implementación del software "DSDE" y se muestran las funcionalidades de Weka usadas en el mismo.

ABSTRACT

The search for drugs against diseases that plague human health remains a priority worldwide. Given the ineffectiveness of the tests of "test and error", it is necessary to find solutions for the development of alternative routes to the discovery of new active compounds. The use of multiclassifiers (classifiers that fuse and the combination of each is obtained as a result) has been a very effective strategy that has allowed a greater accuracy of analysis while producing good results in the search of decision-making. The present work aims to deepen the assembly or multiple classifiers system methods, implementing a system that combines several classification models for better predictions. In combining the outputs of base classifiers, several mathematical functions are used. The selection of base classifiers is done according to a set of statistics that compares the accuracy of prediction and a group of diversity measures that combines the results of individual classifiers and proposes several combinations of them. The main aspects of analysis, design and implementation of software "DSDE" are presented and there are used Weka functionalities in it.

ÍNDICE

ÍNDICE

Lista de Figuras

Lista de Tablas

INTRODUCCIÓN

INTRODUCCIÓN

Las enfermedades parasitarias como malaria (con una incidencia de 500 millones de personas afectadas), enfermedad de Chagas (16 a 20 millones de personas infectadas en el continente Americano), leishmaniasis (más de 12 millones de personas afectadas en el mundo y 350 millones a riesgo de contraerla), trichomonosis (180 millones de personas afectadas en el planeta), entre otras, siguen siendo un azote tanto para los países subdesarrollados como los que están en vías de desarrollo. Incluso algunas escapan al control como la fasciolosis y trichinelosis, constituyendo enfermedades zoonósicas que afectan en España, Inglaterra y otros países desarrollados.

Todo esto hace que la búsqueda de nuevos principios activos para el tratamiento de estas parasitosis se encuentre dentro de las prioridades a nivel mundial. El método tradicional de evaluación de nuevos principios activos basado en el sistema de "prueba y error" a través de ensayos masivos de gran número de sustancias químicas, es cada vez más ineficiente, siendo necesario ensayar más de 10 000 compuestos para encontrar el deseado.

Los métodos *in silico* son una de las pocas técnicas que tienen potencial para mejorar significativamente el descubrimiento y posterior desarrollo de fármacos. En los últimos años, investigadores del Centro de Bioactivos Químicos y de la Facultad de Farmacia de la Universidad Central de las Villas (UCLV), han desarrollado varios modelos de predicción usando el descriptor molecular ***TOMOCOMD CARDD*** y análisis discriminante lineal para identificar nuevos compuestos activos frente a *Trichomonas vaginalis* (Marrero-Ponce, Machado-Tugores et al. 2005; Meneses-Marcel, Marrero-Ponce et al. 2005; Marrero-Ponce, Meneses-Marcel et al. 2006; Marrero-Ponce, Meneses-Marcel et al. 2008; Meneses-Marcel, Rivera-Borroto et al. 2008; Marrero-Ponce, Rivera-Borroto et al. 2009), *Plasmodium spp.*(Montero-Torres, Garcia-Sanchez et al. 2006), *Leishmanias spp.* (Gonzalez-Diaz, Dea-Ayuela et al. 2008) y *Trypanosoma cruzi* (Montero-Torres, Vega et al. 2005; Vega, Montero-Torres et al. 2006).

Para dar continuidad al descubrimiento de nuevos compuestos utilizando grandes bases de datos internacionales, se necesita encontrar aquellos modelos que muestren una

mayor precisión de forma conjunta en la identificación de medicamentos, por lo que se requiere una selección de los modelos que mejoren su poder predictivo.

Para resolver este tipo de problema, donde es precisa la toma de decisiones, hay que tener en cuenta las características, la información disponible y la relación con el conjunto de resultados. La utilización de multiclasificadores es una estrategia, que puede aumentar la precisión de análisis y obtener excelentes resultados en la búsqueda de estas decisiones. Estos multiclasificadores pertenecen a una reciente área de la minería de datos que ha permitido mejorar, en general, la precisión de las predicciones por medio de las combinaciones de clasificadores individuales. Según la bibliografía consultada (Kuncheva 2000), se distinguen unos de otros atendiendo a diversas características: número de clasificadores generales acoplados, tipo de cada clasificador, características de subconjuntos usados, agregación de las decisiones particulares, tipo de función de información, entre otras.

Problema científico

La determinación de los sistemas (modelos) ensamblados idóneos, a partir de un número grande de modelos bases individuales para la predicción de una propiedad o actividad específica, es un problema actual de los estudios de informática-química y biológica, en donde la complejidad de la modelación, el alto costo de experimentación y su implicación ética, al igual que la baja efectividad del uso en cribado virtual de modelos individuales, hace imprescindible su adecuada combinación. En la actualidad no se conoce un ambiente computacional que integre de forma amigable todos los análisis necesarios para descubrir modelos de predicción ensamblados que describan mejor los datos y que ayuden a resolver – o al menos resolver mejor – problemas de predicción de propiedades o actividades de compuestos químicos y biomoléculas.

Objetivo general

Diseñar e implementar una Interfaz Gráfica de Usuario (GUI) que permita el análisis de la diversidad y la selección de modelos bases al igual que el descubrimiento de ***modelos ensamblados*** de predicción para la descripción de propiedades químico-físicas y biológicas de compuestos químicos y biomoléculas.

Para lograr dicho objetivo general, se proponen los objetivos específicos:

1. Seleccionar e implementar parámetros estadísticos, medidas de diversidad pareadas reportadas en la literatura y varias estrategias de combinación de modelos individuales.
2. Diseñar una herramienta computacional que permita explorar el espacio de los modelos ensamblados empleando el enfoque de fusión, con diversas formas de combinación, para definir un modelo ensamblado de predicción que se ajuste mejor a los datos y tenga un mejor desempeño en las predicciones.
3. Implementar todo lo anterior expuesto en una GUI en la plataforma Java 7 que permita, de forma automática y amigable, analizar la diversidad, la selección y descubrimiento de los mejores modelos fusionados.
4. Validar y verificar el desempeño de la herramienta mediante la experimentación con tres problemas de modelación de informática-química: actividad antimalárica, antileishmanial y antiinflamatoria.

Se formularon además las siguientes preguntas de investigación:

1. ¿Cómo determinar los modelos ensamblados idóneos para la predicción de una propiedad o actividad específica de un número grande de modelos bases individuales?
2. ¿Cómo integrar todos los análisis necesarios para descubrir modelos de predicción ensamblados, que describan mejor los datos y que ayuden a resolver problemas de predicción de propiedades o actividades de compuestos químicos y biomoléculas?
3. ¿Cuáles de las medidas de diversidad a estudiar tendrán mayor relevancia para la obtención de mejores resultados?

Justificación de la investigación

Las enfermedades parasitarias siguen siendo un azote tanto para países subdesarrollados como para los que están en vías de desarrollo, por lo que la búsqueda de nuevos principios activos para el tratamiento, se encuentra dentro de las prioridades a nivel mundial. Los métodos *in silico* son una de las pocas técnicas que tienen potencial para mejorar significativamente el descubrimiento y posterior desarrollo de fármacos. Para

dar continuidad al trabajo realizado por investigadores de la UCLV, se necesita encontrar aquellos modelos que muestren una mayor precisión de forma conjunta en la identificación de medicamentos. La utilización de multiclasificadores es una estrategia que puede aumentar la precisión de análisis y obtener excelentes resultados en la búsqueda de estas decisiones.

Viabilidad de la investigación

En la actualidad la construcción de un multiclasificador ofrece una amplia gama de ideas a desarrollar en este trabajo. Para realizar esta investigación se cuenta con la investigación desarrollada por los tutores: Alfredo Meneses Marcel en su tesis de doctorado y Yovany Marrero Ponce relativo al tema de multiclasificadores, así como un poderoso lenguaje de programación: Java.

El primer paso en la realización de este trabajo fue la confección del marco teórico. Para ello se realizó una amplia revisión de la literatura consultando libros, artículos y páginas de internet, entre otras fuentes. Sus elementos esenciales se encuentran expuestos de manera resumida en el primer capítulo de la presente tesis.

Como conclusión de la elaboración del marco teórico se enuncia la siguiente hipótesis de investigación:

El diseño e implementación computacional de pruebas estadísticas, medidas de diversidad, estrategias de combinación de modelos individuales, algoritmos y estrategias de fusión, en un ambiente (GUI) para explorar el espacio de los modelos ensamblados, facilitará encontrar formas de predicción que se ajusten mejor a los datos y ayudar a resolver – o al menos resolver mejor – problemas de quimio-bioinformática.

La tesis está estructurada en tres capítulos. El primero contiene una descripción detallada de los sistemas de múltiples clasificadores. Se definen varias arquitecturas y características así como diferentes estrategias de los métodos de combinación. Además se hace un análisis de diferentes parámetros estadísticos, medidas de diversidad y variantes de combinación para la obtención de los mejores modelos. En el capítulo dos se expone la plataforma de desarrollo y los diagramas creados para las fases de análisis y diseño de la herramienta. Se presentan además las funcionalidades de Weka[1] usadas

[1] WEKA is a Java-written open source. It is available at http://www.cs.waikato.ac.nz/~ml/weka/ under the GNU General Public License.

en la misma. En el capítulo tres se muestra el manual de usuario del software y además se realiza un estudio con tres aplicaciones de datos reales. En estas, se usaron las diferentes funcionalidades que brinda el programa para la identificación de los modelos idóneos ensamblados, optimizando el tiempo y el coste de la investigación. Todos los capítulos terminan con un epígrafe de consideraciones finales en el que se resumen los aspectos más importantes que se trataron. Finalmente se enuncian las conclusiones y recomendaciones, se relaciona la bibliografía y se muestran algunos anexos.

DESCRIPCIÓN DE SISTEMAS DE MÚLTIPLES CLASIFICADORES

CAPÍTULO 1. DESCRIPCIÓN DE SISTEMAS DE MÚLTIPLES CLASIFICADORES

En la literatura internacional se refieren a multiclasificadores como conjuntos de clasificadores diferentes que realizan predicciones que se fusionan, y se obtiene como resultado la combinación de cada una de ellas. El hecho o la idea de combinación hace que los multiclasificadores sean citados a través de distintos términos, entre ellos: métodos de ensamble (Drucker, Cortes et al. 1994; Yan and Goebel 2004); modelos múltiples (multiple models)(Maclin and Shavlik 1995; Smyth 1995; Jacobs 1996); sistemas múltiples de clasificación (multiple classifier systems); combinación de clasificadores (combining classifiers) (Kittler, Hatef et al. 1998); integración de clasificadores (integration of classifiers); mezcla de expertos (mixture of experts)(Jacobs 1996); comité de decisión (decision committee) (Webb 2000); comité de expertos (committee of experts); fusión de clasificadores (classifier fusion) (Cho and Kim 1995; Lynch and Willet 2003; Toh 2004), entre otros.

Los Sistemas de Múltiples Clasificadores (SMC) se han incrementado debido a que resuelven los problemas de sobre adaptación (overfitting) y es posible obtener buenos resultados con pocos datos de entrenamiento, ya que tienen como fundamento principal que son más exactos que los clasificadores individuales, pues la decisión combinada toma ventaja sobre las decisiones individuales de cada clasificador. Así mismo, un problema complejo puede ser descompuesto en varios subproblemas que sean más fáciles de entender y resolver, incluso los errores no correlacionados de los clasificadores individuales pueden eliminarse por medio de la combinación (Sharkey and Sharkey 1995; Kittler, Hatef et al. 1998) .

Existe gran diversidad de diseños de clasificadores. Antiguamente, el objetivo era encontrar el mejor clasificador; actualmente, es sacar el mejor provecho de la gran variedad que existen para obtener mayor eficiencia y precisión. Al poseer un rendimiento elevado y cierto grado de infinidad se puede afirmar, que si el número de clasificadores aumenta, la probabilidad de errores disminuye; aunque esto trae consigo que algunos métodos de combinación entrenados alcancen resultados inconsistentes y/o que ganen una complejidad computacional inaceptable.

Para construir multiclasificadores se necesitan dos etapas:

- La primera, teniendo en cuenta la aplicación específica, considera qué tipo y cuántos multiclasificadores son necesarios para analizar un problema determinado, y realizar el diseño de los mismos.
- La segunda etapa trata el problema de cómo combinar los resultados obtenidos por los multiclasificadores para obtener el resultado más óptimo.

La efectividad del sistema multiclasificador estará en dependencia de que ambas etapas de diseño, sean concebidas en forma conjunta. (L. Xu, Krzyzak et al. 1992).

1.1 Aprendizaje automático

El Aprendizaje Automatizado es una rama de la Inteligencia Artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender, o sea crear programas que puedan, en un sentido similar a lo realizado por los humanos, aprender por sí mismos. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. (Bello García 2012).

El aprendizaje automático permite una gran variedad de enfoques y técnicas, y puede aplicarse a diferentes problemas de clasificación. A continuación se muestran dos tipos de enfoques de aprendizaje dependiendo de la base de conocimiento: supervisado y no supervisado.

1.1.1 Aprendizaje supervisado

Los algoritmos basados en aprendizaje supervisado, producen una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Para este tipo se distinguen perfectamente rasgos predictores o rasgos de entrada y los rasgos objetivos o de salida, respondiendo estos a una etiqueta. Un ejemplo de este tipo de algoritmo es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar (clasificar) una serie de vectores utilizando una entre varias categorías (clases). La base de conocimiento del sistema está formada por ejemplos etiquetados. Este tipo de aprendizaje puede llegar a ser muy útil en problemas de investigación biológica, biología computacional y Bioinformática. (Bello García 2012).

Diferentes métodos se han empleado para dar solución al problema de clasificación entre los cuales aparecen: k-Vecinos más Cercanos, Redes Bayesianas, Árboles de Decisión, Redes Neuronales Artificiales, entre otros.

1.1.2 Aprendizaje no supervisado

En este tipo de problemas todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos, o sea la base de conocimiento del sistema está formada por ejemplos no etiquetados. Por lo tanto, en este caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas. Una forma de aprendizaje no supervisado es la agrupación o *clustering*. (Bello García 2012)

Los algoritmos no-supervisados más comunes son Cobweb (Fisher 1987), k-Means (MacQueen 1967), entre otros.

1.2 Arquitectura

Si se dispone de un conjunto de clasificadores, la arquitectura dependerá de la forma en que se desea integrar el conjunto de estos para garantizar una toma de decisión, esto es, que sean independientes unos de otros, por eliminación de hipótesis (decisiones dependientes) o a través de la cooperación de clasificadores (cada uno soluciona un problema). (Last, Bunke et al. 2002; Rahman and Fairhurst 2003).

1.2.1 Arquitectura vertical o secuencial

En la arquitectura vertical o secuencial, la información resultante de un clasificador se convierte en información de entrada para otro (Figura 1.1).
La organización en niveles sucesivos de decisión permite reducir progresivamente el número de clases posibles. En cada nivel: un único clasificador tiene en cuenta la respuesta proporcionada por el clasificador colocado anteriormente para tratar los rechazos y confirmar la decisión obtenida en el eslabón anterior.

Existen dos enfoques: la re-evaluación y la reducción del conjunto de clases (L. Xu, Krzyzak et al. 1992).

En el primer caso, el principio fundamental es asegurar la re-evaluación de los ejemplos que se rechazan o se reconocen con muy baja confianza en cualquier nivel.

Los datos de entrada se presentan al primer clasificador en la configuración. El primer experto genera un valor de confianza que se corresponde al indicador de decisión. La confianza puede ser un valor:

- Superior al umbral establecido y la decisión es la aceptación, la re-evaluación en el siguiente nivel es innecesaria.
- Que se encuentre dentro del rango definido por el primero y el segundo umbral, en este caso dependiendo de las restricciones se toma una decisión u otra.
- Inferior al segundo umbral y la decisión es el rechazo, lo que hace necesaria la re-evaluación en el nivel posterior.

En las situaciones donde se necesita la re-evaluación, el clasificador siguiente realiza la búsqueda de una solución en todo dominio de clases; genera otro valor de confianza que indica la posible clase y este proceso se repite para los sucesivos clasificadores hasta que un clasificador encuentre un valor de confianza suficientemente alto o el clasificador final emite su decisión.

En el segundo caso, el clasificador primario en la configuración genera una lista de posibilidades que constituye un subconjunto del número total de clases. El próximo clasificador limita su análisis al subconjunto de clases generado por el clasificador anterior y así sucesivamente.

La adopción de esta arquitectura está acompañada de un filtrado progresivo de las decisiones (reducción de la ambigüedad) y es sensible al orden en el cual se colocan los clasificadores, por lo que se debe tener un conocimiento *a priori* del comportamiento de cada uno de los clasificadores. De manera general, es difícil de optimizar el conjunto ya que existe dependencia.

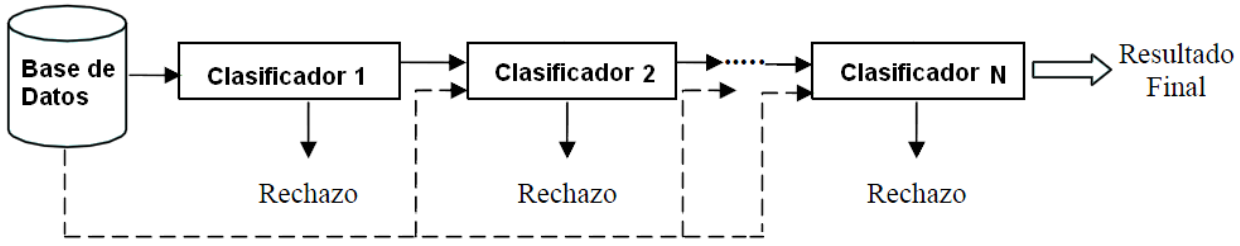

Figura 1. 1 Arquitectura vertical o secuencial con el enfoque de re-evaluación

1.2.2 Arquitectura horizontal o paralela

En la arquitectura paralela, los clasificadores operan independientemente unos de otros, luego se fusionan sus respectivas respuestas y se busca un acuerdo entre los clasificadores para llegar a una única decisión (Figura 1.2).

El esquema es muy fácil de aplicar ya que no requiere de una reparametrización de los otros clasificadores en caso de que existan modificaciones en el conjunto, aunque la activación de todos los clasificadores conlleva a un costoso tiempo de cálculo.

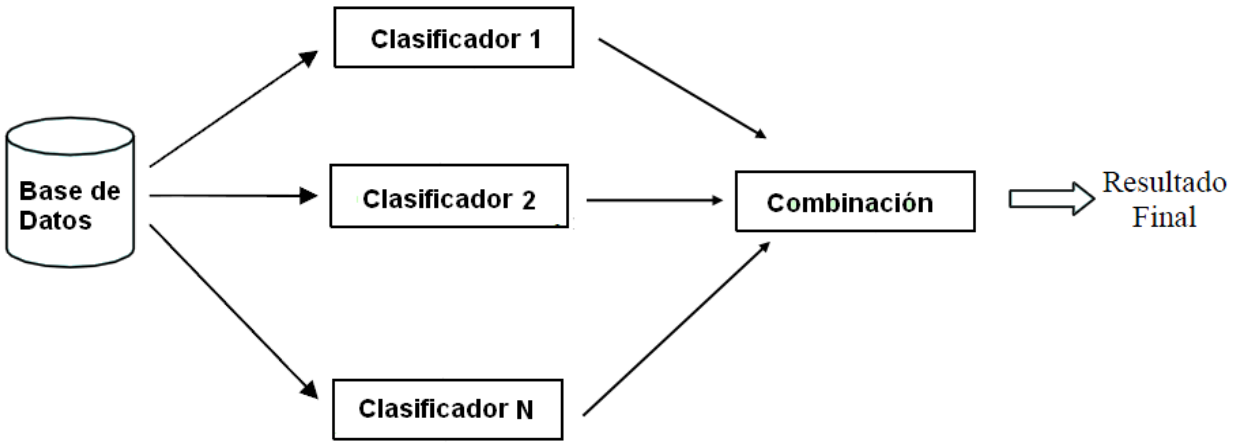

Figura 1. 2 Arquitectura horizontal o paralela

1.2.3 Arquitectura híbrida

En la arquitectura híbrida, se combinan las ventajas de las dos anteriores, o sea, la reducción del conjunto de clases posibles y la búsqueda de un consenso entre los clasificadores. Con la unión de estas, se obtiene un mejor provecho de cada uno de los clasificadores que se utilicen (Figura 1.3).

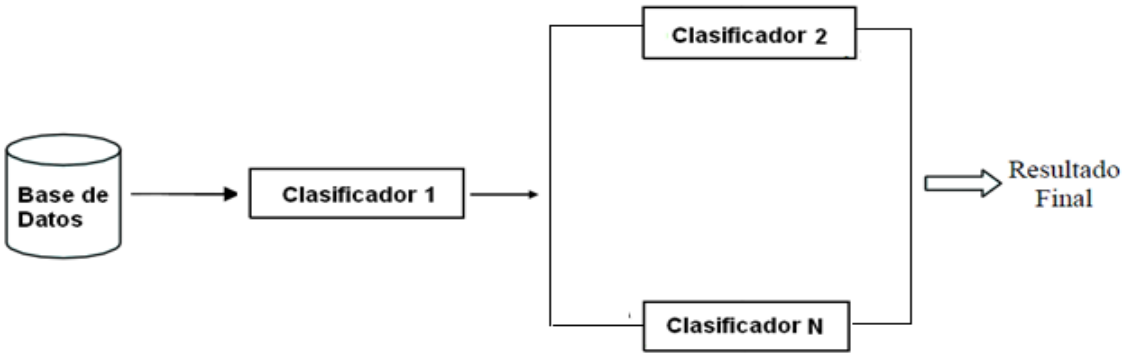

Figura 1. 3 Arquitectura híbrida

1.3 Características de los métodos de ensamble

Existen muchas razones del porqué combinar clasificadores. Cada método de clasificación se basa en conceptos o procedimientos de estimación diferentes, tratando de aunar las mejores propiedades de cada uno de ellos combinándolos de alguna forma. La combinación de estos muestra mayor precisión que cualquiera de ellos de manera individual.

Hansen y Salomon (1990) establecen la precisión y la diversidad como requisitos necesarios y suficientes para llevar a cabo con éxito la combinación de dos o más sistemas de clasificación (Hansen and Salomon 1990). Ellos probaron que si la tasa de error media para una observación es menor del 50% y los clasificadores utilizados son independientes en la producción de errores, el error esperado para una observación puede ser reducido a cero cuando el número de clasificadores combinado se acerca a infinito. Incluso, el método de voto mayoritario, que es el más sencillo de combinación, y bajo el supuesto de que los clasificadores son independientes entre sí, se puede comprobar para un problema dicotómico cómo la precisión del conjunto es superior a la de los clasificadores individuales, siempre que estos cometan un error inferior al 50% (un clasificador se considera preciso si su error es inferior a 0.5, dos clasificadores individuales son diversos cuando sus errores de salida no son correlacionados).

Krogh y Vedelsby (1995) probaron después, que el error conjunto puede dividirse en un término que mide el error de generalización medio de cada clasificador individual y un término que recoge el desacuerdo entre los clasificadores, o sea, que la combinación ideal consiste en clasificadores con alta precisión que estén el mayor número de veces posible en desacuerdo (Krogh and Vedelsby 1995).

Dietterich (2000) plantea tres razones para justificar la superioridad de la combinación de clasificadores sobre los individuales: razón estadística, razón computacional y razón representacional (Dietterich 2000).

- **Razón estadística**

Un sistema de aprendizaje puede verse como la búsqueda dentro de un determinado espacio de hipótesis. Escoger un clasificador que resuelva el problema entraña un riesgo, ya que este puede no ser el más capacitado para resolver el caso que se está tratando, o que el conjunto de datos posea un tamaño demasiado pequeño en

comparación con el tamaño del espacio de hipótesis. Debido a esto, puede que la combinación no nos proporcione mejores resultados que el mejor clasificador individual que se dispone, pero sí que elimina o mitiga el riesgo de equivocarnos en la selección.

En la Figura 1.4, f representa el mejor clasificador, H es el espacio de hipótesis, la región central contiene los clasificadores que clasifican bien en el problema a tratar (Dietterich 2000).

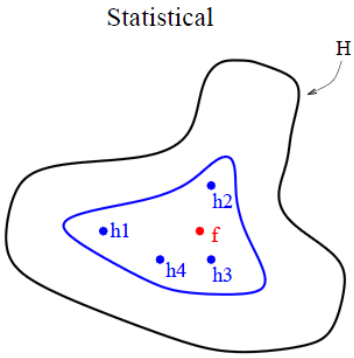

Figura 1. 4 Razón estadística para combinación de clasificadores

- **Razón Computacional**

Aún teniendo un volumen de datos que haga desaparecer el problema estadístico mencionado anteriormente, existe el problema de determinados sistemas de clasificación que realizan algún tipo de búsqueda local por lo que pueden quedar atrapados en un óptimo local. La combinación de clasificadores obtenidos realizando la búsqueda local desde puntos iniciales distintos, logrará una mejor aproximación a la función objetivo buscada a diferencia de lo que consiguen acercarse estos clasificadores individualmente.

En la Figura 1.5, f representa el mejor clasificador, H es el espacio de hipótesis y las líneas discontinuas muestran la trayectoria hipotética de los clasificadores durante su entrenamiento (Dietterich 2000).

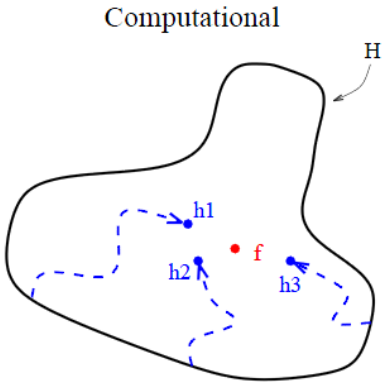

Figura 1. 5 Razón computacional para combinar clasificadores

- **Razón representacional**

Se presenta cuando el espacio de búsqueda no contiene ninguna solución que sea una buena aproximación a la función inicial. Mediante combinaciones de hipótesis relativamente sencillas se puede llegar a una mejor aproximación, ya que la combinación puede darnos la oportunidad de expandir el espacio de las funciones que pueden ser representadas, con la posibilidad que esto acarrea de que consigamos incluir el objetivo.

En la Figura 1.6, f representa el mejor clasificador y H es el espacio de hipótesis (Dietterich 2000).

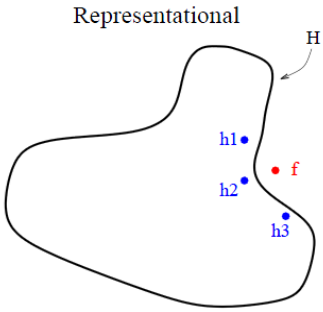

Figura 1. 6 Razón representacional para combinar clasificadores

1.4 Toma de decisiones. Estrategias de Combinación

Para realizar la combinación de las decisiones individuales de los clasificadores, se proponen en la literatura varias estrategias de combinación, siendo las más usadas: selección y fusión de clasificadores (Kuncheva 2000; Bulacio 2006). Este trabajo estará basado en la estrategia de fusión.

1.4.1 Fusión de clasificadores

La fusión de clasificadores asume que todos los clasificadores son competitivos y complementarios (igualmente *expertos*). Por este motivo, cada uno de ellos emite una decisión respecto a cada patrón de prueba que se presenta. Esta se puede aplicar siempre que exista redundancia en el análisis, o sea, si hay más de un clasificador capaz de evaluar el problema (Bulacio 2006).

1.4.2 Otras estrategias de Combinación

Segrera y Moreno proporcionan tres estrategias de combinación, a pesar de que las salidas de los clasificadores individuales puedan ser diferentes: métodos de nivel abstracto, métodos de nivel de rango y métodos de nivel de medidas (Segrera and Moreno 2006).

1.5 Taxonomía de los métodos de Combinación

En las últimas décadas se han propuesto varias clasificaciones de los métodos de combinación de los sistemas de aprendizaje. El estudio de los mismos puede hacerse desde distintos puntos de vista según las características que presente.

1.5.1 Clasificación según Masulli y Valentini

Masulli y Valentini utilizan una clasificación en función de si el algoritmo actúa o no sobre los clasificadores básicos modificándolos. Esta clasificación se dividen en dos métodos de combinación: generadores y no generadores (Valentini and Masulli 2002).

1.5.1.1 Métodos generadores

Los métodos generadores crean conjuntos de clasificadores básicos actuando sobre el propio sistema de clasificación o sobre el conjunto de datos de entrenamiento. Estos intentan mejorar la precisión global de la combinación mediante la actuación directa sobre la precisión y diversidad de los clasificadores base. Se pueden distinguir diferentes métodos en función de las estrategias que utilizan para conseguir mejorar los clasificadores básicos:

- Métodos de selección de variables
- Métodos de prueba y selección

- Métodos aleatorios de agregación
- Métodos de remuestreo

1.5.1.2 Métodos no generadores

Los métodos no generadores se restringen a combinar un conjunto dado de clasificadores básicos, o sea, intentan combinar de la mejor manera los ya existentes. Los clasificadores básicos se ensamblan mediante un procedimiento de combinación en dependencia de su capacidad de adaptación a las observaciones de entrada y las necesidades de la salida que facilitan los sistemas de aprendizaje individuales (el tipo de combinación depende del tipo de salida).

Si solo se dispusiera de la clase asignada o si las salidas continuas son difíciles de manipular, se utilizaría el voto mayoritario (forma más sencilla de combinar multiclasificadores). Si los clasificadores básicos proporcionan las probabilidades a *posteriori* de las clases se pueden agregar operadores sencillos como el mínimo, máximo, media, mediana, producto o media ponderada.

1.5.2 Clasificación según Bulacio

Bulacio los clasifica teniendo en cuenta el tipo de análisis de los clasificadores que forman el sistema: especializados (multiclasificadores de especialistas) o no especializados (multiclasificadores de generalistas) (Bulacio 2006).

1.5.2.1 Métodos de especialistas

Los clasificadores especializados o de especialistas tienen una cobertura parcial del problema que tratan. Estos se subdividen en dependencia de si los operadores realizan tareas de selección estática (también llamada por Bulacio como agregación por síntesis) o selección dinámica (agregación por selección) de clasificadores.

1.5.2.2 Métodos de generalistas

Los clasificadores no especializados o de generalistas tienen una visión completa del problema que tratan. La subdivisión es de acuerdo a si existe o no aprendizaje del método de combinación (fusión simple o fusión entrenada). Esta estrategia emplea clasificadores capaces de analizar individualmente el problema completo. La efectividad de realizar una combinación dentro de sistemas con redundancia estará ligada a la

correlación precisión-diversidad de los clasificadores y a la capacidad del combinador de alcanzar un beneficio del trabajo colectivo.

1.5.2.2.1 Fusión simple

No poseen entrenamiento en la etapa de agregación (combinación de varios clasificadores) de resultados individuales. La elección de un método simple depende de las características de los clasificadores.

1.5.2.2.2 Fusión entrenada

El salto cualitativo en el desarrollo de los sistemas de combinación en paralelo fue la inclusión de conceptos comunes al trabajo en grupo referidos a otros ámbitos. La consideración de esta información en la obtención de la decisión colectiva hizo posible mejorar la precisión en aplicaciones reales, aunque se hace necesaria una etapa más de entrenamiento.

1.5.3 Otras Clasificaciones

Kittler y colaboradores, (Kittler, Hatef et al. 1996; Kittler, Hatef et al. 1998; Kittler and Alkoot 2003) consideran relevante la representación de la información de entrada al sistema, distingue según el punto de vista de análisis, básicamente dos escenarios de combinación. En el primero, todos los clasificadores utilizan la misma representación en los patrones de entrada u observaciones. En el segundo caso, cada clasificador utiliza su propia representación de las observaciones de entrada.

Bezdek y colaboradores, (Bezdek, Pal et al. 1999) los analizan según el tipo de fusión de información: fusión de datos, fusión de características o fusión de clasificadores.

Dietterich diferencia entre los métodos de combinación aquellos que realizan una votación de tipo Bayesiano, los que modifican los ejemplos de entrenamiento, los que modifican las variables, los que modifican las clases posibles, y por último los que aleatorizan el sistema de aprendizaje (Dietterich 2000).

Lam propone agrupar estos métodos según la arquitectura de la agregación distinguiendo entre si esta se realiza en serie, en paralelo o de forma jerárquica (Lam 2000).

Jan y col. recogen la separación en función de si los clasificadores básicos son seleccionados o no por el algoritmo de combinación, diferenciando entre los métodos de

combinación orientados a la selección y aquellos orientados a la combinación (JAIN and DUIN 2000).

Kamely y Wanas tienen en cuenta el tipo de dependencia del algoritmo de combinación con los datos de entrenamiento, pudiendo ser implícita, explícita o independiente (Kamel and Wanas 2003).

1.6 Métodos de Combinación

Algunos de los métodos de combinación descritos en la literatura internacional son los siguientes:

- Voto mayoritario
- Promedio
- Producto
- Máximo
- Mínimo
- Mediana
- Cuentas de Borda
- Cuentas de Borda por peso
- Integrales Borrosas

1.6.1 Nomenclatura

Consideremos un sistema formado por n clasificadores, $X = \{X_1,...,X_n\}$ y K ejemplos disponibles para el aprendizaje. Cada X_i (con i = 1,…, n) tipifica una muestra s según el conjunto de alternativas posibles $W = \{w_1, ...,w_c\}$.

La ejecución del proceso de clasificación conjunta es realizada en dos etapas. En la primera es donde se obtienen las decisiones individuales y la segunda es donde se realiza la combinación de dichas decisiones. Los elementos que intervienen en el proceso son los siguientes:

- Primera etapa: Cada X_i asigna c valores, a través de su función de clasificación $f(X_i(s))$, que relacionan a la entrada s con el espacio de alternativas. En forma resumida, $f_i = [0, 1]^c$, donde $[0, 1]^c$ es el vector normalizado de decisión individual o evidencia.

- Segunda etapa: La información de partida es el conjunto de las *n* decisiones individuales asociadas a los *n* clasificadores (f_1,...,f_n) junto con el conocimiento que tenga el sistema para realizar la combinación.

Las conclusiones provistas por los clasificadores se organizan dentro de lo que denominamos perfil de decisiones (*DP*) para facilitar su posterior procesamiento.

$$DP(s) = \begin{bmatrix} f_1^1 & \cdots & f_1^c \\ \vdots & \ddots & \vdots \\ f_n^1 & \cdots & f_n^c \end{bmatrix}$$

Cada fila se corresponde con la salida de un clasificador: f_i= {f_i^1, ..., f_i^c}; siendo f_i^j la decisión del clasificador X_i respecto a la clase W_j . Cada columna es el resultado de todos los clasificadores en relación a una clase: $X(j)$= {f_1^j, ..., f_n^j}; es el conjunto de decisiones de los *n* clasificadores sobre la clase W_j . Por último, a la decisión conjunta del sistema sobre todas las clases la denominamos y_G. (Bulacio 2006).

1.6.2 Métodos de Combinación implementados

1.6.2.1 Voto mayoritario

La técnica de Voto Mayoritario funciona de forma similar a como el ser humano realiza el proceso de elecciones políticas. La votación por mayoría, es considerada una regla que interpreta cada resultado de clasificación como voto para una de las clases de datos y asigna el patrón de la entrada a la clase que recibe mayoría de votos (Giacinto, Roli et al. 2000).

Esta se divide en: Voto Mayoritario Simple y Voto Mayoritario Ponderado

1.6.2.2 Promedio

Este operador realiza el promedio en cada una de las clases sobre todo el conjunto de clasificadores. La salida global es la clase que alcanza el mayor valor promedio.

$$y_G = \arg max_{j=1}^{c} \left(\frac{1}{n}\sum_{i=1}^{n} f_i^j\right)$$

1.6.2.3 Producto

Si los vectores de las características usados por los clasificadores individuales son distintos, se puede establecer la regla del producto como la multiplicación de las probabilidades de una misma clase en todos los clasificadores, que sería:

$$\alpha = \prod_{j=1}^{m} V_j$$

1.6.2.4 Máximo

Para cada patrón *j* se calcula el máximo valor de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el mayor valor. La base de clasificadores debe proporcionar decisiones con medidas numéricas. (Kittler, Hatef et al. 2002).

$$\psi_G = \arg max_{j=1}^{c} \left(max\left(f_1^i, \ldots, f_n^j\right)\right) \forall nasignada$$

1.6.2.5 Mínimo

Para cada patrón *j* se calcula el mínimo valor de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el menor valor. La base de clasificadores debe proporcionar decisiones con medidas numéricas.

$$\psi_G = \arg min_{j=1}^{c} \left(min\left(f_1^i, \ldots, f_n^j\right)\right) \forall nasignada$$

1.6.2.6 Mediana

Para cada patrón *j* se calcula el valor que es la mediana de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el máximo valor. (Kittler, Hatef et al. 2002).

$$\psi_G = \arg max_{j=1}^{c} \left(med\left(f_1^i, \ldots, f_n^j\right)\right) \forall nasignada$$

1.7 Caracterización de la diversidad. Medidas de diversidad

Autores como Masulli y Valentini, (Valentini and Masulli 2002) llegan al criterio de que la clasificación conjunta puede ser mejor que la individual si existe diversidad y precisión, procurando tener pocos errores y diversos, pero no existe un consenso en cómo medirla.

La caracterización de diversidad en la capacidad de generalización puede verse en dos tipos: cualitativa y cuantitativa. La primera es útil para elegir el conjunto de selección-combinación de individuos, mientras que la segunda pondera la diversidad del grupo a través de medidas que pueden ser incluidas dentro de los procesos del multiclasificador.

1.7.1 Análisis cualitativo

Con este análisis se puede determinar la variabilidad del análisis del conjunto. Cuando se utiliza un conjunto de clasificadores dados, el análisis cualitativo es el punto de partida en la determinación de características relevantes en los clasificadores considerados.

1.7.2 Análisis cuantitativo

En este análisis se cuantifican los aspectos más relevantes a la precisión del proceso de combinación, valorando distintos aspectos relacionados con el comportamiento colectivo a través de medidas.

Kuncheva y col. (2002) (Kuncheva and Shipp 2002) y Kuncheva y Whitaker (2002) (Kuncheva and Whitaker 2002) plantean que no hay una medida de diversidad involucrada en forma explícita en los métodos de generación de clasificadores, aunque asumen que la diversidad es el punto clave en cualquiera de los métodos. Ellos estudian la correlación de medidas entre pares (X_i, X_j) o grupos de clasificadores y la precisión del sistema, a través de una recopilación de distintas medidas propuestas por diversos autores para valorar diversidad, concluyendo que tan sólo las medidas de doble falta y la medida de dificultad están asociadas a la precisión colectiva. La elección de la medida a utilizar va a depender directamente de la cantidad de clasificadores seleccionados.

Algunas de las propiedades que deben cumplir las medidas de diversidad son las siguientes:

> **Propiedad 1:** La medida de diversidad tiene un valor finito.
>
> **Propiedad 2:** La medida de diversidad está acotada, cuentan con un valor mínimo y un valor máximo para su adecuada interpretación.
>
> **Propiedad 3:** El resultado de la medida de diversidad puede ser representada en forma vectorial.

Propiedad 4: La medida de diversidad es simétrica. Una medida de diversidad puede ser simétrica o no con respecto a la correcta o incorrecta clasificación (0 ó 1). El cumplimiento de esta propiedad no se considera ventajoso o perjudicial, simplemente es una característica propia de la medida de diversidad.

Las medidas pueden ser clasificadas como: medidas en forma de pares (pairwise) y medidas para todo el conjunto (non pairwise).

Las medidas en forma de pares se calculan por pares de clasificadores usando sus salidas, las cuales son binarias (0,1) que indica si la instancia fue correctamente clasificada o no por el clasificador. Algunas de ellas son: desacuerdo, doble falta, Q estadístico, entre otras.

A continuación se indica el resultado de dos clasificadores (C_i, C_j) para una instancia en cuanto si la clasificaron correctamente o no.

Tabla 1. 1 Tabla de clasificación entre los resultados de los clasificadores C_i y C_j para una instancia

	C_j correcto (1)	C_j incorrecto
C_i correcto (1)	*a*	*b*
C_i incorrecto	*c*	*d*
a + b + c + d = 1		

Si se suman para todas las instancias los valores de a, b, c, d entre el par de clasificadores (C_i, C_j) se obtendrá el siguiente resultado, a partir del cual se calculan las medidas en forma de pares:

Tabla 1. 2 Tabla de clasificación entre los resultados de los clasificadores C_i y C_j para todas las instancias

	C_j correcto (1)	C_j incorrecto
C_i correcto (1)	*A*	*B*
C_i incorrecto	*C*	*D*
A +B + C + D = N		

Donde *A* sería igual a la suma total de los valores de *a* para todas las instancias y así respectivamente con los valores de *B, C y D.* El número total de casos es *N*. Un conjunto de *L* clasificadores produce *L (L – 1)/2* pares de valores. Para obtener un único resultado habría que promediar estos valores.

Mientras que las medidas de diversidad que se basan en todo el conjunto consideran a todos los clasificadores a la vez y calculan un único valor de diversidad para todo el conjunto. Algunas de estas medidas son: varianza de Kohavi-Wolpert (Kohavi and Wolpert 1996), información mutua (Valentini and Masulli 2002), diversidad generalizada (Krzanowski and Partridge 1995), entre otras.

1.7.3 Medidas de diversidad entre pares de clasificadores (pairwise)

1.7.3.1 Medida de desacuerdo

Usada por Skalak (Skalak 1996) y Ho T. Kam (1998) (Kam 1998) para caracterizar la complementariedad entre clasificadores; evalúa la cantidad de ejemplos en los que ha habido desacuerdo sobre el conjunto total de muestras de prueba. Mientras mayor sea su valor mayor será la diversidad.

$$D_{ij} = \frac{B + C}{N}$$

$D_{ij} \in$ [0,1]

1.7.3.2 Medida de doble falta

Otra de las medidas que se analizará se conoce como medida de doble falta introducida por Giacinto y Roli (Giacinto and Roli 2001), considera el fallo de los dos clasificadores al mismo tiempo. Ruta y Gabrys(Ruta and Gabrys 2001) definen a esta medida como una medida no-simétrica. Esto quiere decir que si se intercambian los unos con los ceros en los resultados de los clasificadores, el valor de la medida no va a ser el mismo. Esta medida está basada en el concepto de que es más importante conocer cuántos errores simultáneos son cometidos, que cuándo ambos tienen clasificación correcta. Mientras menor sea el valor mayor será la diversidad.

$$DF_{ij} = \frac{D}{N}$$

$DF_{ij} \in$ [0,1]

1.7.3.3 Diferencia entre medida de desacuerdo y medida de doble falta

Es la resta entre la medida de desacuerdo y la medida de doble falta de los dos clasificadores. Mientras mayor sea el valor de esta medida mayor será la diversidad entre los clasificadores.

$$DI_{ij} = D - DF$$

$DI_{ij} \in [0,1]$

1.7.3.4 Coeficiente de correlación

El coeficiente de correlación permite medir el grado de concordancia en las decisiones de los clasificadores, o sea, el grado de relación que las decisiones individuales tienen entre sí (Kuncheva and Shipp 2002; Kuncheva and Whitaker 2002):

$$\rho_{ij} = \frac{A * D - B * C}{\sqrt{(A + B) * (C + D) * (A + C) * (B + D)}}$$

El coeficiente de correlación cuenta con las siguientes características:

a) El valor del coeficiente de correlación varía entre [-1,1]. Ambos extremos representan relaciones perfectas entre las decisiones y 0 representa la ausencia de asociación (los valores más pequeños indican mayor grado de diversidad).
b) Cuanto más cercano sea el coeficiente de correlación al valor -1, más débil será la asociación entre los clasificadores.
c) Una relación positiva significa que los clasificadores que obtienen calificaciones altas en una variable tienden a obtener calificaciones altas en la otra. Una relación negativa se presenta cuando los clasificadores que obtienen calificación baja en una variable tienden a obtener calificación baja en la otra.

1.7.3.5 Q-statistic de Yule

Estadístico usado para evaluar similitud entre clasificadores (Yule 1990):

$$Q_{ij} = \frac{A * D - B * C}{A * D + B * C}$$

$Q \in$ [-1,1]

Q= 0 para clasificadores estadísticamente independientes

Q> 0 para clasificadores que tienden a reconocer las mismas muestras

Q< 0 cuando los errores se dan en distintos ejemplos

1.8 Evaluación en la clasificación

No existe un modelo de clasificador mejor que otro de manera general; para cada problema nuevo es necesario determinar con cuál se pueden obtener mejores resultados, y es por esto que han surgido varias medidas para evaluar la clasificación y comparar los modelos empleados para un problema determinado. Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión que se obtiene cuando se prueba el clasificador en un conjunto de datos que no intervienen en el entrenamiento (Bonet Cruz 2008). A continuación se muestra la matriz de confusión de un problema de dos clases, donde C1 es la clase negativa y C2 la clase positiva:

Tabla 1. 3 Matriz de confusión entre clase real y clase obtenida

	Clase obtenida	
Clase Real	C1	C2
C1	TN	FP
C2	FN	TP

Leyenda:

TP y TN: cantidad de elementos bien clasificados de la clase positiva y negativa, respectivamente.

FP y FN: cantidad de elementos negativos y positivos mal clasificados, respectivamente.

Basados en estas medidas, se calcula el coeficiente de correlación de Matthews (Matthews correlation coefficient) (Baldi, Brunak et al. 2000), la exactitud (accuracy) (GONELL 2010), la sensibilidad (sensitivity), la especificidad (specificity), y la razón de FP (FAR), que se dan por las expresiones siguientes:

1.8.1 Coeficiente de correlación de Matthews

El coeficiente de correlación es siempre entre -1 y 1 y se puede utilizar con variables no binarios. Es una medida para variables normalizadas y tienden a tener la misma magnitud y signo. Un valor de -1 indica total desacuerdo y 1 totalmente de acuerdo. El coeficiente de correlación es 0 para las predicciones completamente al azar.

$$CCM = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FN) * (TP + FP) * (TN + FP) * (TN + FN)}}$$

1.8.2 Exactitud

La exactitud mide la proporción de patrones que han sido bien predichos en relación al número total de éstos. Esta medida da una idea de la fiabilidad del sistema y está íntimamente ligada al error ya que ambos valores suman 1.

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

1.8.3 Sensibilidad

En sistemas de clasificación, para la clase i, la sensibilidad es la proporción de patrones predichos positivamente que son correctamente identificados.

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

1.8.4 Especificidad

En sistemas de clasificación, la precisión de la clase i es la proporción de patrones predichos positivamente que son correctamente clasificados.

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

1.8.5 Razón de falsos positivos o Razón de falsa alarma (FAR)

En sistemas de clasificación, la FAR es la proporción entre los falsos positivos y el total de valores negativos.

$$FAR = \frac{FP}{FP + TN}$$

1.9 Consideraciones finales del capítulo

En estudios realizados hasta la actualidad, no se conoce un ambiente computacional que integre de forma amigable todos los análisis necesarios para descubrir modelos de predicción ensamblados que describan mejor los datos y que ayuden a resolver problemas de predicción de propiedades o actividades de compuestos químicos y biomoléculas. Debido a ello se hace imprescindible la adecuada combinación de los modelos individuales, de manera que se minimice la complejidad de la modelación, el alto costo de experimentación y su implicación ética así como la baja efectividad del uso en cribado virtual.

En el siguiente capítulo se propone un modelo de multiclasificador el cual utiliza diferentes clasificadores de base cada uno de los cuales será entrenado con diferentes subconjuntos de rasgos, previamente seleccionados. Se propone además una aplicación para calcular la diversidad entre grupos de clasificadores, la cual será de mucha utilidad para conformar los modelos de multiclasificadores.

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE "DESDE 1.0"

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE "DSDE 1.0"

Este capítulo alude a las generalidades del análisis, diseño e implementación del software "DSDE" (Diversity analysis, Selection and Discovery of best Ensemble from bases models) versión 1.0. Se expone la plataforma de desarrollo y los diagramas creados para las fases de análisis y diseño de la herramienta. Además, se ofrece una breve explicación de la implementación, lo cual facilita su comprensión y extensión.

2.1 Análisis, diseño e implementación de la herramienta

El lenguaje UML (Unified Modeling Language) (Rumbaugh and Booch 2000) se utilizó para el diseño de la herramienta DSDE versión 1.0. Este tiene como objetivos principales la especificación, visualización, construcción y documentación de los productos de un sistema de software. Este lenguaje es usado por el RUP (Rational Unified Process) (Jacobson, Booch et al. 2000) como lenguaje de modelado para lo cual se basa en todos sus tipos de diagramas, que constituyen diferentes vistas del modelo del producto. La siguiente figura ilustra los diagramas que componen la estructura de un producto escrito por el lenguaje UML:

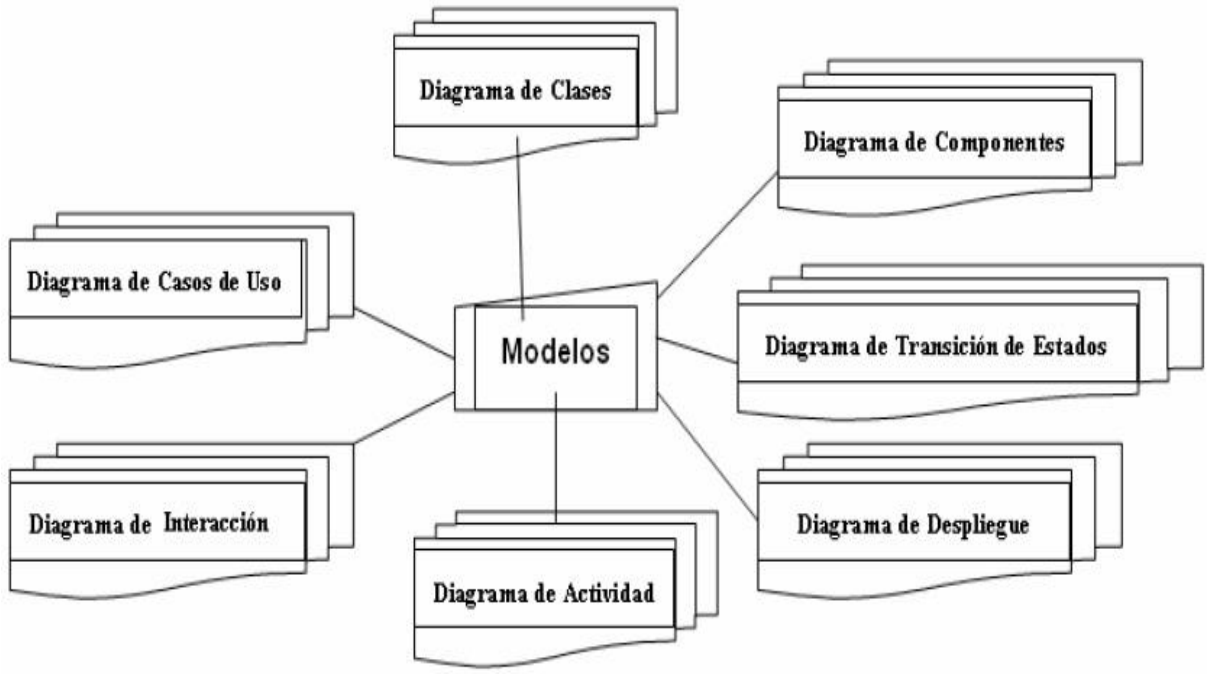

Figura 2.1 Diagrama UML

De los diagramas UML que muestra la figura anterior, empleamos: Diagrama de Casos de Uso, Diagrama de Actividad y Diagrama de Clases.

La herramienta empleada para el modelado de todos los diagramas correspondientes a las fases de análisis y diseño fue *Visual Paradigm para UML versión 9.0*.

2.1.1 Diagrama de casos de uso

Los modelos de casos de uso proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales del sistema basándose en los requerimientos de los usuarios. Ellos dirigen todo el proceso de desarrollo de un software ya que constituyen el punto de partida para llevar a cabo la mayoría de las actividades: el análisis, diseño y prueba del software (Jacobson, Booch et al. 2000). Este modelo se realiza identificando cada actor del sistema como los posibles usuarios para los cuales está realizado el mismo.

La herramienta **"**DSDE**"** está destinada a cualquier tipo de usuario, pudiendo ser un estudiante, especialista o investigador en química, biología, computación o ramas similares. En el diagrama de la Figura 2.2 se le ha nombrado a ese actor como Usuario.

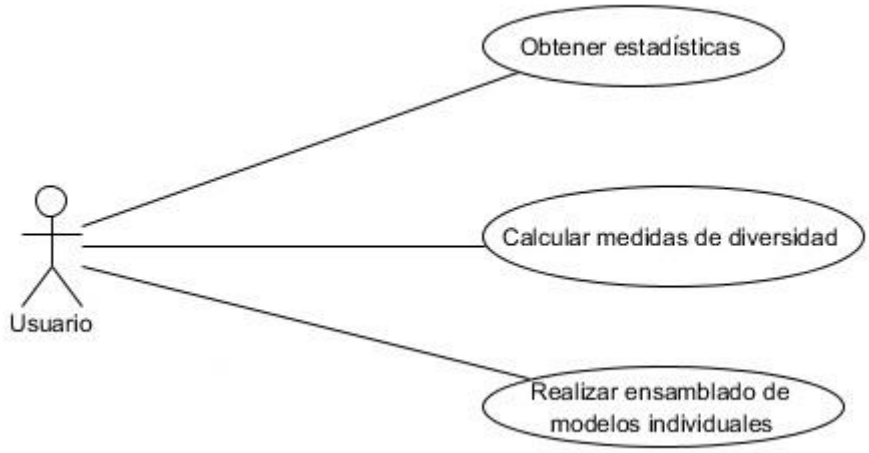

Figura 2.2 Diagrama de casos de uso

El usuario mediante el primer caso de uso puede obtener estadísticas como coeficiente de correlación de Mathews, exactitud, especificidad, sensibilidad y razón de falsa alarma. El actor mediante el segundo caso de uso puede obtener los mejores modelos a combinar a partir de diferentes medidas de diversidad que le permitan resolver problemas de diversas áreas de aplicación. El usuario utiliza el tercer caso de uso para

realizar funciones matemáticas como voto mayoritario, promedio, producto, máximo, mínimo y mediana lo que permite obtener los mejores modelos ensamblados.

2.1.2 Diagrama de clases

La técnica del diagrama de clase se ha vuelto medular en los métodos orientados a objetos. El diagrama de clase describe los tipos de objetos que hay en un sistema y las diversas clases de relaciones estáticas (asociaciones, subtipos) que existen entre ellos. También muestra los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos (Fowler and Scott 1997).

El software se compone de la interfaz de usuario con nombre *DSDE* que contiene las funcionalidades propias del sistema. Además se hace uso de la biblioteca new-weka-paralell.jar para cargar y manejar los datos.

Weka es una plataforma de aprendizaje implementada en Java. Esta herramienta sigue los preceptos del código abierto (open source), por lo que su código fuente está totalmente disponible, permitiendo la modificación del mismo. Solo es necesario recompilarlo para posteriormente agregar extensiones al sistema. Además es un software de distribución gratuita lo que posibilita su uso, copia, estudio, modificación y redistribución sin restricciones de licencias.

En la Figura 2.3 se muestran las clases y paquetes más importantes que representan las operaciones que se realizan, con sus atributos y métodos más relevantes mediante un diagrama de clases en UML.

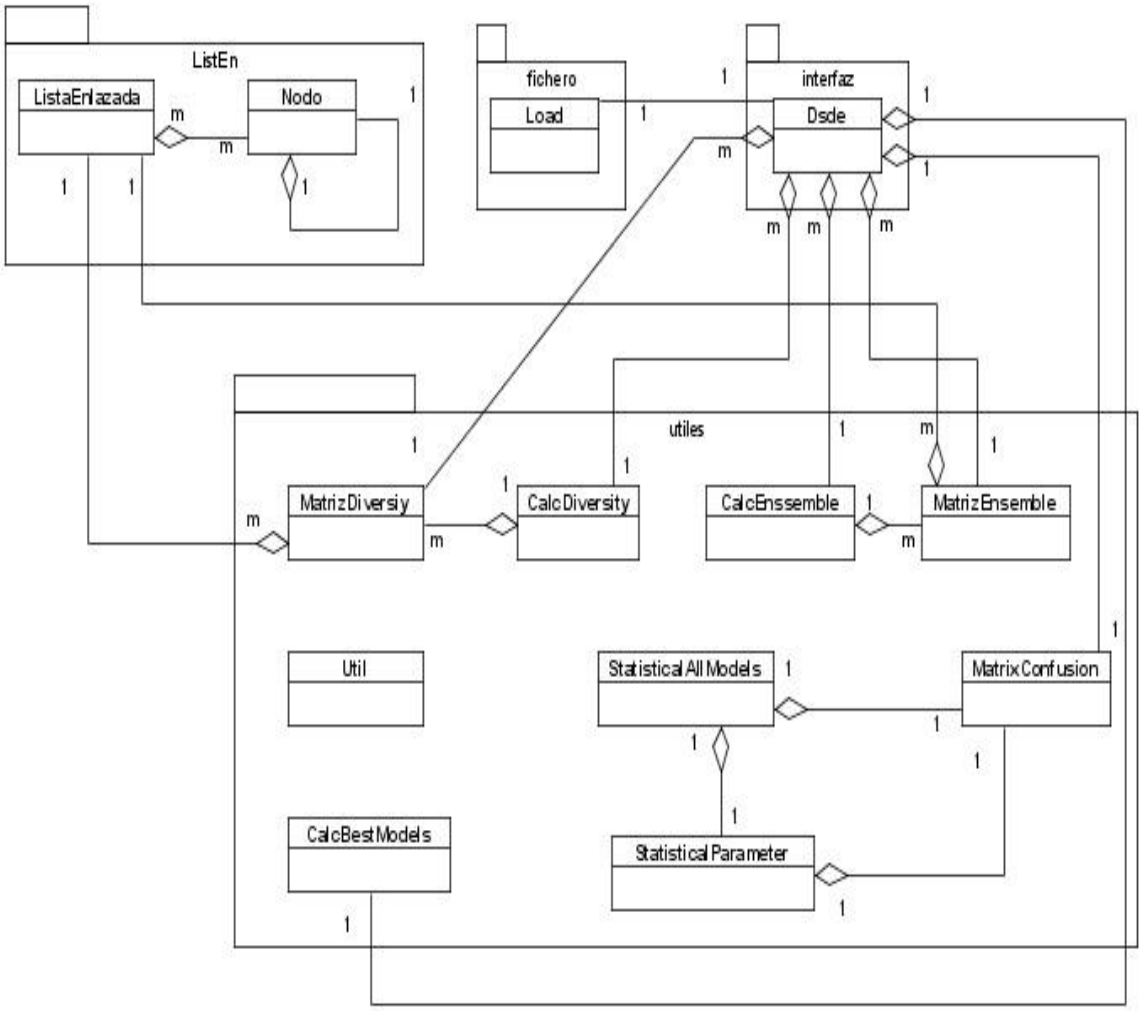

Figura 2.3 Diagrama correspondiente a las clases y paquetes más importantes

La clase *MatrizDiversity* almacena información referente a los clasificadores, es la encargada de calcular el valor de las matrices *A, B, C, D* y devolver en una lista este valor de la matriz correspondiente a cada combinación.

La clase *CalcDiversity* es una de las más importantes, ya que calcula el valor de las medidas de diversidad y obtiene los *m* modelos de multiclasificadores más diversos de acuerdo a cada una de estas medidas, donde el número *m* es un parámetro indicado por el usuario.

La clase *MatrizEnsemble* contiene la implementación de las diferentes funciones de combinación, además tiene un método donde se construye el nombre de los nuevos multiclasificadores.

Otra de las clases más importantes es *CalcEnsemble* ya que esta es la encargada de construir la data de salida. Entre sus principales atributos están una lista con los valores asociados a las reglas de combinación, un arreglo con los nombres y una data de tipo *Instances* la cual se llenará con el resultado final del ensamble, el método *makeHeader(n,t)* es el encargado de llenar la data con los nombres y valores.

La clase *Util* contiene la implementación de un conjunto de métodos que son utilizados en diferentes clases.

La clase *MatrixConfusion* tiene como atributos un arreglo con la cantidad de casos bien y mal clasificados, un arreglo con el total de elementos correspondiente a cada clase y la data de entrada. Esta clase tiene un método que es el encargado de calcular los valores asociados a la matriz de confusión.

La clase *CalcBestModels* obtiene los mejores modelos ensamblados dependiendo de la opción seleccionada por el usuario.

La clase *StatisticalParameter* calcula las estadísticas asociadas a cada modelo individual construyendo la matriz de confusión para cada uno. El resultado lo deposita en un arreglo de tipo double.

La clase *StatisticalAllModels* almacena la información estadística obtenida a partir de la clase *StatisticalParameter* asociada a cada modelo individual.

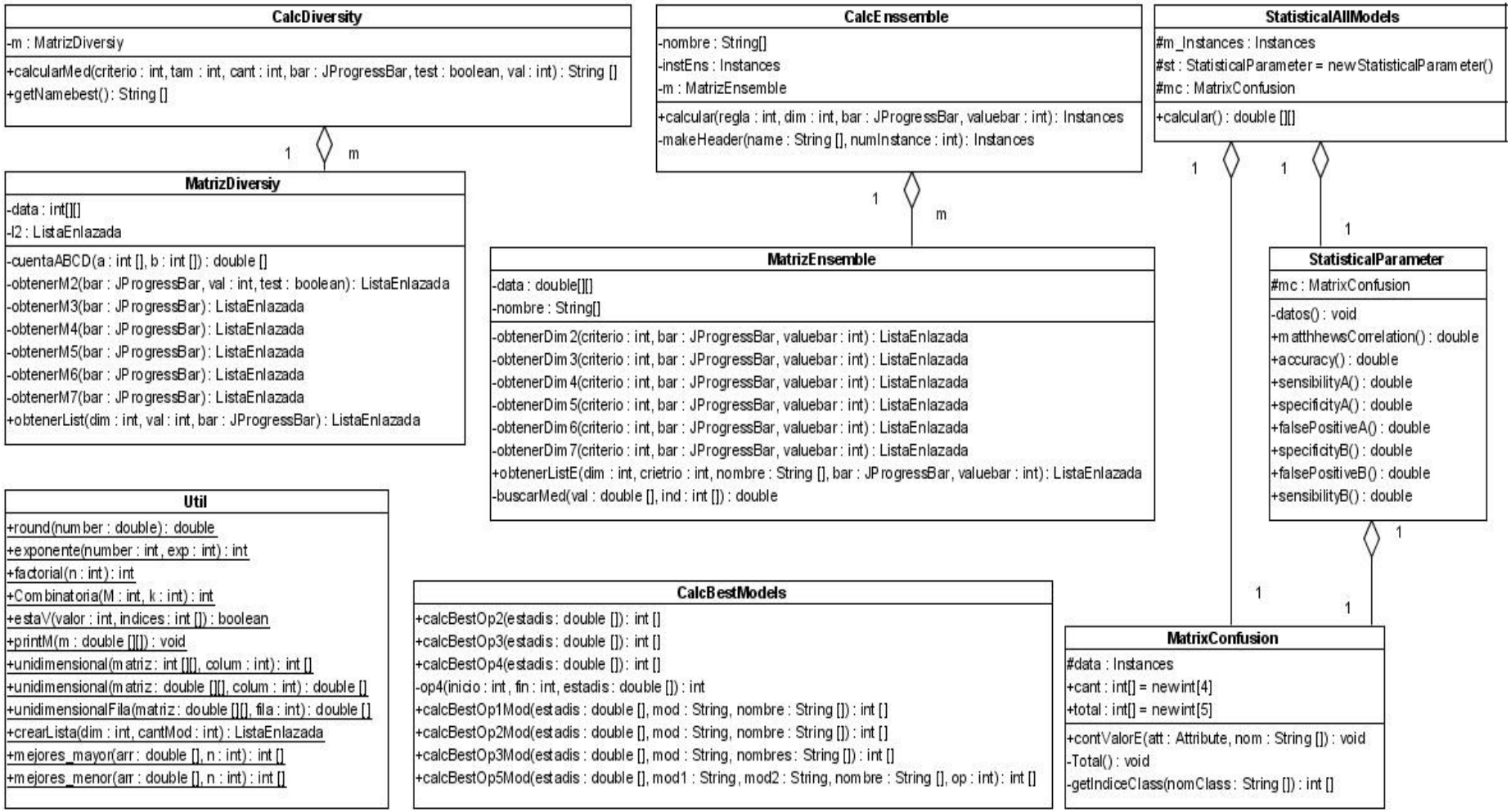

Figura 2.4 Diagrama de clases correspondiente al paquete *utiles*

La clase *Load* es la encargada de leer los diferentes tipos de ficheros de entrada como txt, csv y arff, haciendo uso de algunas clases de Weka.

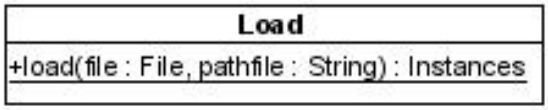

Figura 2.5 Diagrama de clases correspondiente al paquete fichero

La clase *ListaEnlazada* construye la lista con la información referente a las medidas de diversidad y a las funciones de combinación respecto a los datos, la cantidad de nodos de esta depende de la cantidad de combinaciones que se realicen.

La clase Nodo es la que almacena la combinación entre los modelos, la matriz A, B, C, D para calcular medidas de diversidad y modelos ensamblados a medida que se realizan las operaciones.

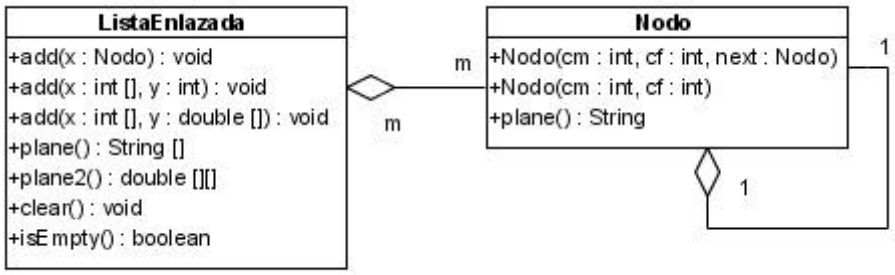

Figura 2.6 Diagrama de clases correspondiente al paquete ListEn

Dsde es la clase principal en la cual se utilizan todas funcionalidades implementadas en las clases anteriores.

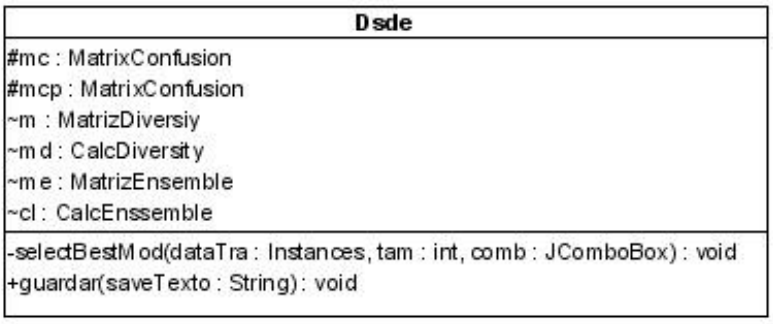

Figura 2.7 Diagrama de clases correspondiente al paquete interfaz

2.1.3 Diagrama de actividad

Los diagramas de actividad se utilizan para modelar los aspectos dinámicos de un sistema, lo que generalmente implica modelar los pasos secuenciales (y posiblemente concurrentes) de un proceso computacional.

Las figuras 2.8, 2.9 y 2.10 muestran tres diagramas de actividades para los casos de uso “Obtener estadísticas”, “Calcular medidas de diversidad” y “Realizar ensamblado”, donde pueden apreciarse a través de modelos simples la descripción del flujo de actividades asociada a cada caso de uso.

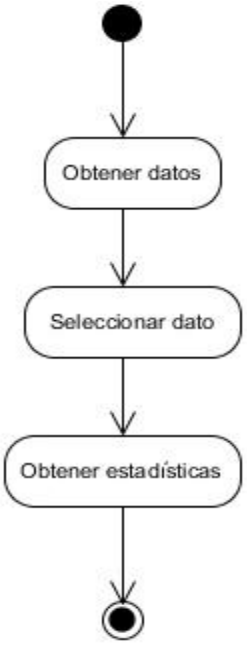

Figura 2.8 Diagrama de actividad correspondiente al caso de uso: Obtener estadísticas

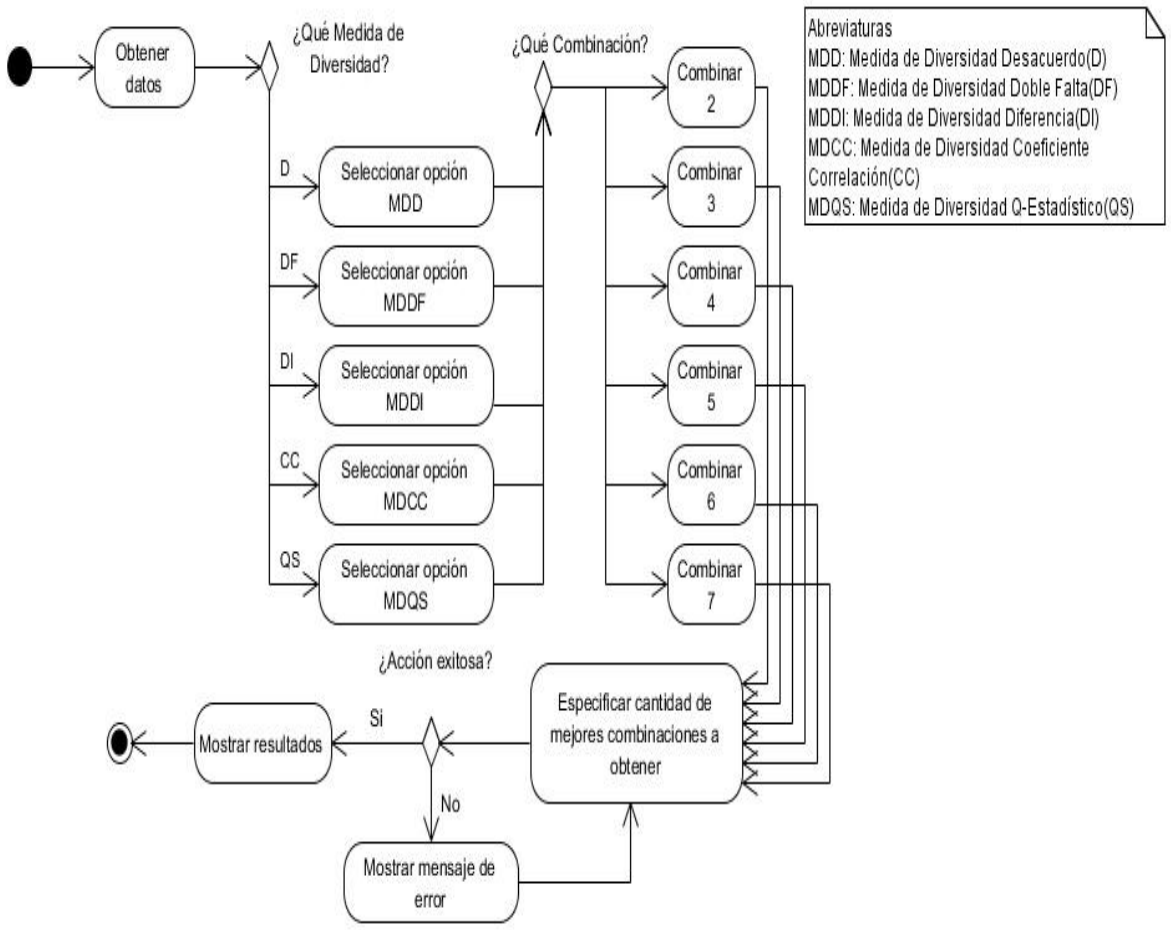

Figura 2.9 Diagrama de actividad correspondiente al caso de uso: Calcular medidas de diversidad

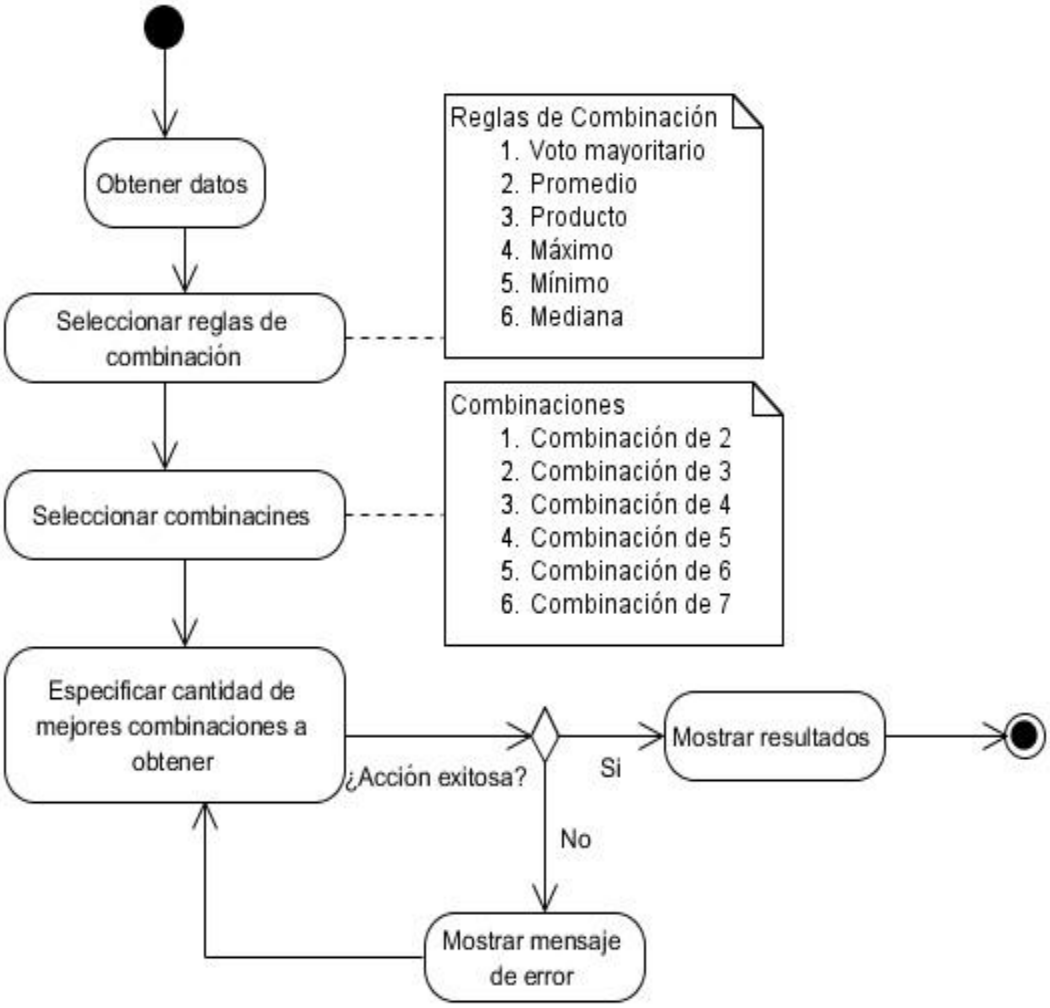

Figura 2.10 Diagrama de actividad correspondiente al caso de uso: Realizar ensamblado de modelos individuales

2.1.4 ¿Cómo agregar una nueva medida de diversidad?

Las medidas de diversidad utilizadas en el software *DSDE* no son las únicas que existen, de hecho en la literatura existen otras, pero no se ha probado todavía la superioridad de ninguna. En este trabajo hemos implementado las medidas en forma de pares más utilizadas y de mejor comprensión al usuario. El sistema está diseñado de forma que el usuario puede agregar una nueva medida de diversidad teniendo en cuenta algunas consideraciones.

Si se desea agregar una nueva medida en forma de pares, habría que agregarla en el método *calcularMed()* de la clase *CalcDiversity*. Las medidas en forma de pares se calculan en función de los parámetros *a*, *b*, *c*, y *d* que son valores en las matrices *A*, *B*, *C*, y *D* para un par de clasificadores.

A continuación se muestra cómo se agrega una nueva medida de diversidad en forma de pares.

```
Switch(med) {
    Case <número>:
        Switch(dim) {
            (…)
        }
    Break;
}
```

Donde cada caso del primer switch representa las diferentes medidas de diversidad, por lo que el número de la nueva medida debe ser consecutivo a los anteriores. Cada caso del segundo switch representa las combinaciones que se pueden realizar, es igual que en las medidas anteriores lo único que cambia en el código es la forma de calcular la medida, el valor se guarda en el arreglo *result*.

Si se desea agregar una medida en forma de no pares habría que hacer un nuevo método en la clase *CalcDiversity* e implementarla, ya que en nuestro trabajo no hay ninguna.

2.1.5 ¿Cómo adicionar un nuevo método de combinación?

En la literatura estudiada aparece una gran cantidad de métodos de combinación, los métodos de combinación implementados en nuestro trabajo son los que aparecen en el software Weka ya que son los más usados y aportan buenos resultados.

Para agregar una nueva regla habría que hacerlo en la clase *MatrizEmsemble* en la cual hay un método para cada tamaño de combinación.

A continuación se muestra cómo agregar una nueva regla de combinación.

```
Switch (metodo){
    Case <número>:
    Break;
}
```

Donde cada caso del switch representa un método de combinación y el número debe ser consecutivo a los anteriores. De igual forma se haría para cada método de la clase dependiendo del tamaño de las combinaciones.

2.2 Funcionalidades de Weka usadas en la herramienta

Las clases de Weka están organizadas en diferentes paquetes. Un paquete es la agrupación de clases e interfaces donde lo habitual es que las clases que lo formen estén relacionadas y se ubiquen en un mismo directorio. Esta organización de la estructura de Weka hace que añadir, eliminar o modificar elementos no sea una tarea compleja. Este software está formado por 10 paquetes globales, y dentro de ellos se agrupan otros paquetes que aunque su contenido se ajusta al paquete padre, ayudan a organizar aun mejor la estructura de clases e interfaces.

Para nuestra herramienta usamos dos de los paquetes globales: *core* y *filters* (Figura 2.11). Del paquete *core* utilizamos las clases *Attribute*, *Instance* e *Instances*. Del paquete *filters* utilizamos la clase *Remove* y modificamos la clase *NumericToNominal*, de forma tal que se le pasa como parámetro el índice del atributo que se quiere convertir. Estos filtros pertenecen a la categoría de filtros no supervisados de atributos.

Una de las clases que es necesario tener en cuenta en nuestro modelo es la clase *Attribute*, que representa a un rasgo o descriptor, entre sus atributos podemos destacar, el nombre, el tipo y el índice que ocupa. Los métodos más usados son:

- Los métodos *isNominal*, *isNumeric*, *isRelationValued*, *isString*, *isDate* retornan verdadero si el atributo es del tipo especificado en el nombre del método
- *name*: retorna el nombre del atributo
- *numValues*: retorna el número de valores del atributo. Si este no es nominal, cadena o relación de valores retorna cero

La clase I*nstance* también es de gran importancia, ya que representa un caso o instancia de la base de datos, y contiene entre otros atributos el valor de cada uno de los rasgos para este caso. Tiene una serie de métodos entre los cuales están:

- *value()*: devuelve el valor del atributo pasado como parámetro
- *classAttribute*: devuelve la clase de la que procede el atributo

- *classValue*: devuelve el valor de la clase del atributo

La última de las clases usadas de este paquete es I*nstances* que representa a la base de datos completa. Tiene entre sus atributos el nombre de la base, la lista de atributos y la lista de casos. Entre sus métodos se encuentran:

- *numAttributes()* y *numInstances()* los cuales devuelven el número de atributos e instancias respectivamente de la base de datos
- *Attribute()*: existen dos métodos con este nombre, la diferencia es que uno recibe como parámetro el índice del atributo, y el otro el nombre, ambos retornan el atributo
- *classIndex()*: devuelve el índice del atributo clase

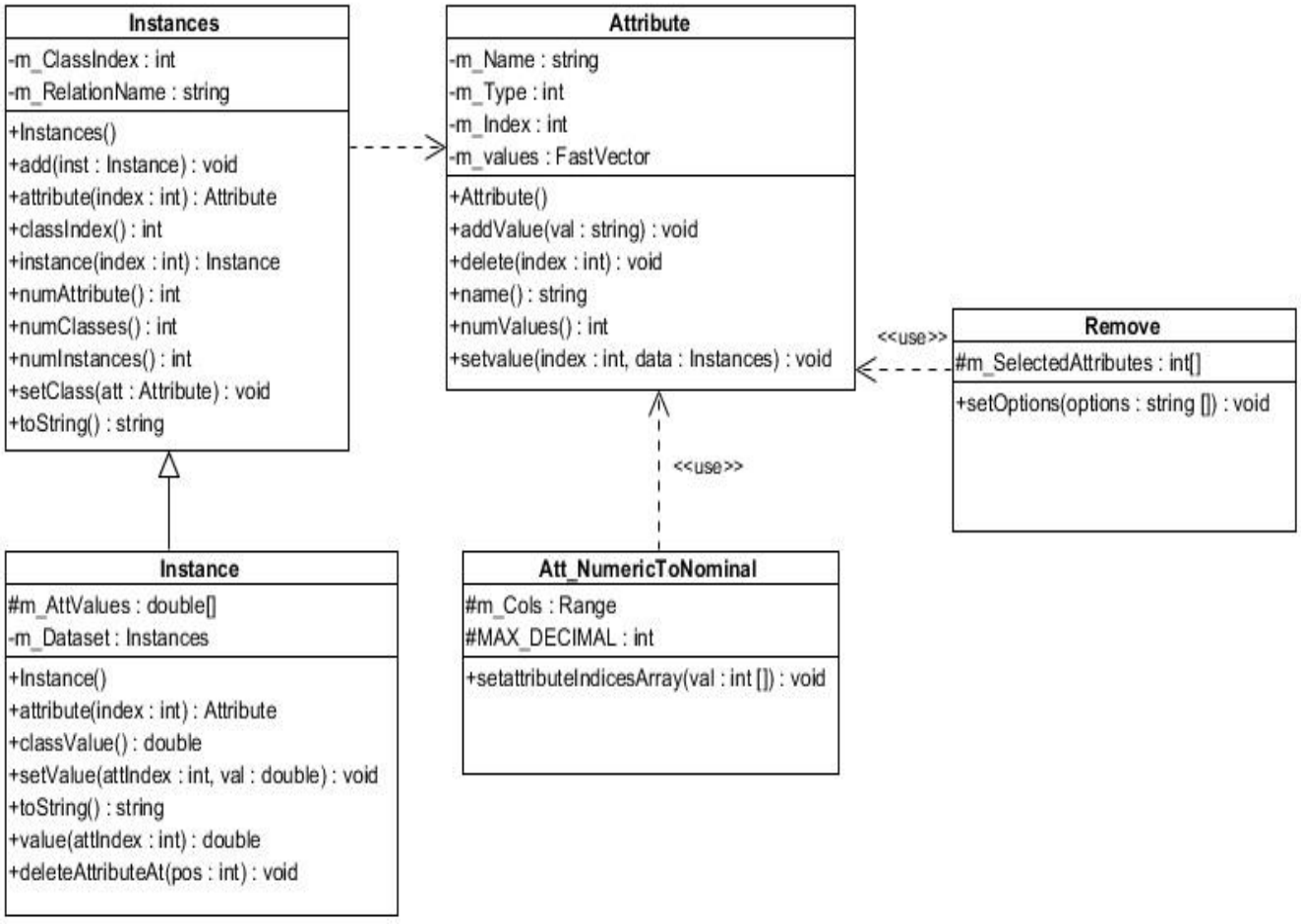

Figura 2.11 Diagrama de clases utilizadas de Weka

2.3 Consideraciones finales del capítulo

Se diseñó e implementó una herramienta que permite la construcción de un multiclasificador el cual utiliza varios modelos de clasificadores de base. Esta herramienta contiene diferentes estadísticas, medidas para obtener la diversidad existente entre un conjunto de clasificadores. Combina las salidas de los clasificadores de base con seis funciones matemáticas: promedio, producto, mínimo, máximo, mediana y voto mayoritario. Esta aplicación resulta de ayuda para la elección de los mejores clasificadores de base que formarán el nuevo multiclasificador.

MANUAL DE USUARIO

CAPÍTULO 3. MANUAL DE USUARIO

En el presente capítulo se realiza una presentación al usuario de las facilidades y funcionalidades del software "DSDE 1.0" de tipo *wizard*. Se efectúa un análisis de las opciones y modo de uso de cada una de ellas, lo cual constituye una sencilla pero valiosa guía para el trabajo con el software. Posteriormente se muestran aplicaciones con datos reales para valorar el nivel de satisfacción de los investigadores del Centro de Bioactivos Químicos (CBQ) y el grupo *Unit of Computer-Aided Molecular Biosilico Discovery and Bioinformatic Research* (CAMD-BIR Unit) de la facultad de Química-Farmacia. En estas aplicaciones se obtienen nuevas combinaciones de modelos individuales de manera que se pueda optimizar el uso para el descubrimiento de nuevos compuestos.

3.1 Manual de usuario

El software "DSDE" versión 1.0 de tipo *wizard* proporciona un sistema capaz de encontrar los modelos que mejor combinan. El usuario puede obtener parámetros estadísticos como coeficiente de correlación de Matthews, exactitud, especificidad, sensibilidad y razón de falsos positivos. Además permite realizar cálculos de medidas de diversidad tales como desacuerdo, doble falta, diferencia entre desacuerdo y doble falta, coeficiente de correlación y Q-estadístico. Esta herramienta desarrollada en Java también facilita la obtención de funciones de combinación como voto mayoritario, promedio, producto, máximo, mínimo y mediana.

3.1.1 Requerimientos

La aplicación requiere tener instalada la Máquina Virtual de Java.

3.1.2 Fichero de entrada

El fichero de entrada del programa puede tener extensión .txt, .csv o .arff, en el caso de los ficheros .txt los datos deben estar separados por tabs y en los ficheros .csv y .arff deben estar separadas por coma.

A continuación se especifica el formato del fichero con extensión .arff.

El formato del encabezamiento es el siguiente:

@relation <nombre-de-la-relación>, donde: <nombre-de-la-relación> es de tipo cadena.

La sintaxis para declarar los atributos es:

@attribute <nombre-del-atributo> <tipo>, donde: <nombre-del-atributo> es de tipo cadena y <tipo> puede ser numérico o nominal.

La sección de datos se encabeza con @data <conjunto-de-datos>, donde en <conjunto-de-datos> se especifican todas las instancias; separando los valores de los atributos para una misma instancia entre comas y las instancias (relaciones entre los atributos) con saltos de línea.

En el caso de que algún dato sea desconocido se expresará con un símbolo "?" o un espacio en blanco.

Los datos que se entren en cualquiera de estos formatos deben tener en la primera columna los nombres de los casos que se corresponde al nombre de los compuestos, la segunda corresponde a la Especificación de Introducción Lineal Molecular Simplificado o SMILES, que se refiere a una notación lineal para codificar estructuras moleculares (en caso de no existir estos valores debe aparecer vacía), a partir de la tercera columna se ponen los clasificadores individuales, (deben aparecer como mínimo tres clasificadores), la penúltima columna corresponde a la clase (valor real) y en la última la data a la que pertenece (entrenamiento o predicción).

3.1.3 Ventana inicial del software

Al ejecutar el software "DSDE" primeramente aparece el *splash* durante unos pocos segundos (Figura 3.1).

Figura 3.1 *Splash* del software DSDE v1.0

3.1.4 Primer paso: cargar datos (“*Load Dataset*”)

La Figura 3.2 muestra el primer paso de la aplicación. Esta interfaz presenta en la parte izquierda de la ventana los diferentes pasos de la aplicación, en la parte superior aparece un panel donde se cargan los datos y se muestra información de los mismos. La Figura 3.3 muestra la ventana que aparece al dar click en el botón “Browse”.

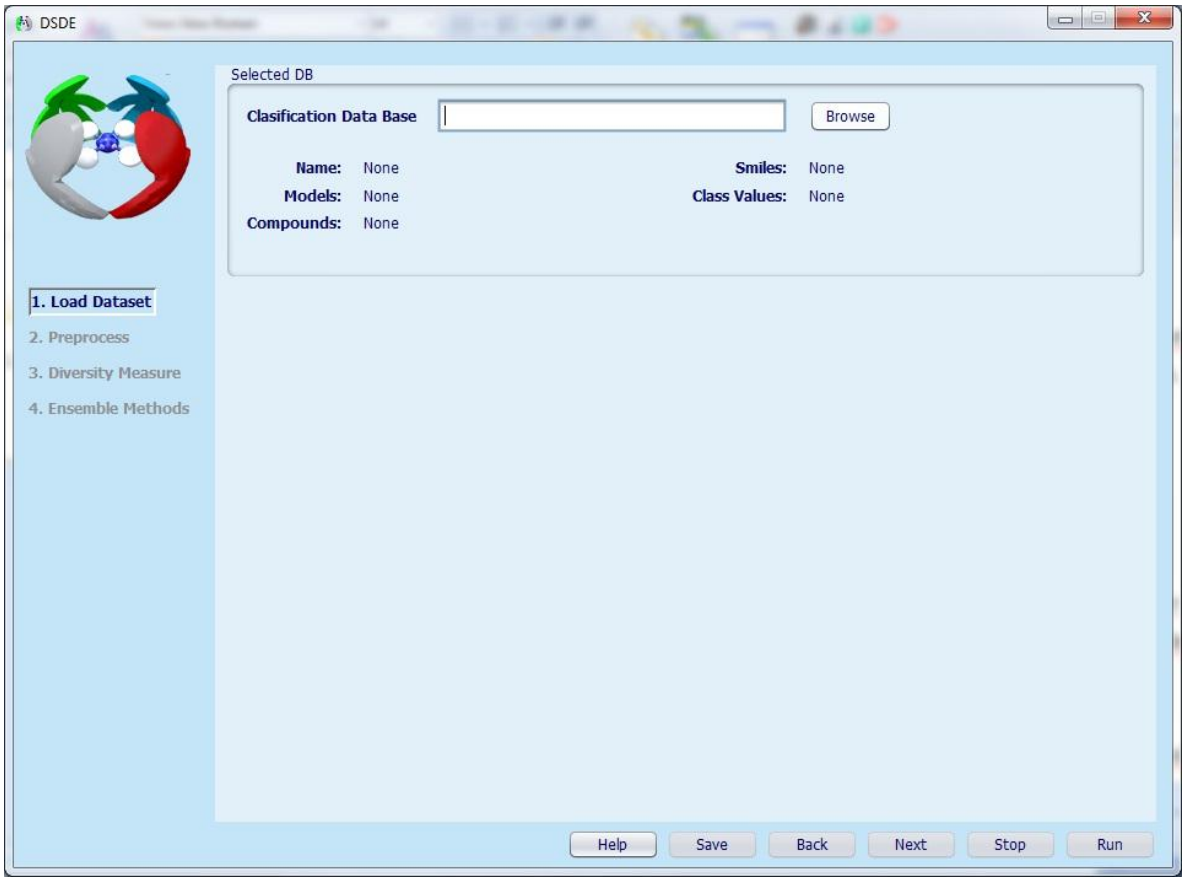

Figura 3.2 Primer paso: Ventana inicial.

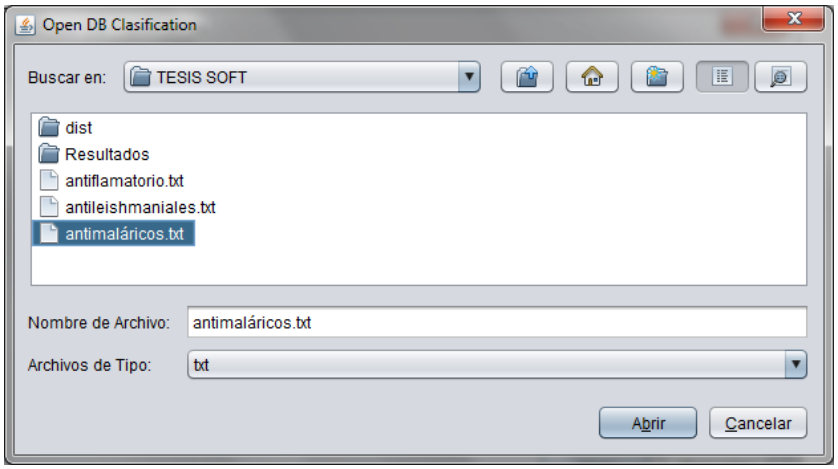

Figura 3.3 Ventana de cargar datos

Una vez cargado los datos, aparece en la parte inferior un panel donde se selecciona la data usada para entrenamiento y la(s) data(s) usada(s) para predicción, también se muestra una vista de la base de datos.

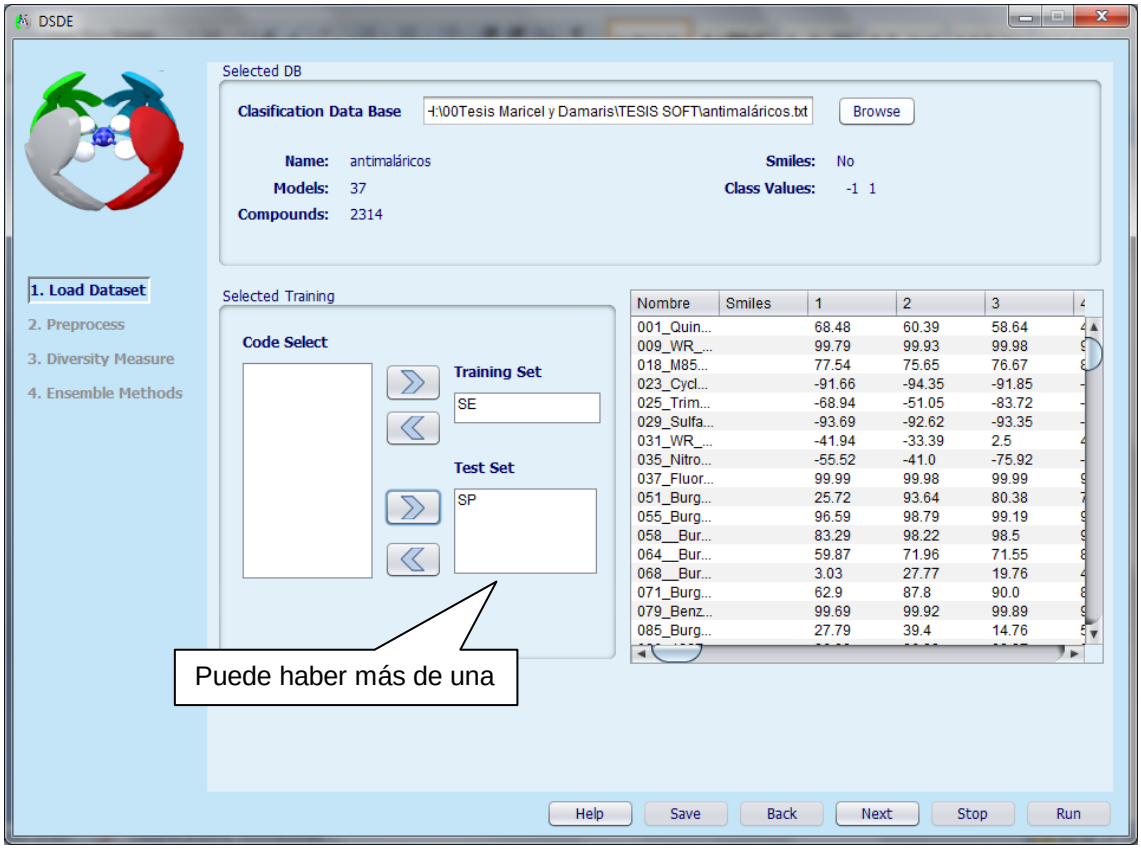

Figura 3.4 Información de los datos.

Entre los pasos 1 y 2 existe una ventana intermedia para seleccionar el tipo de dato con el que se está trabajando (P = Probabilidad y ΔP = Diferencia de probabilidades) y los valores que pertenecen a cada una de las clases como se muestra en la siguiente figura.

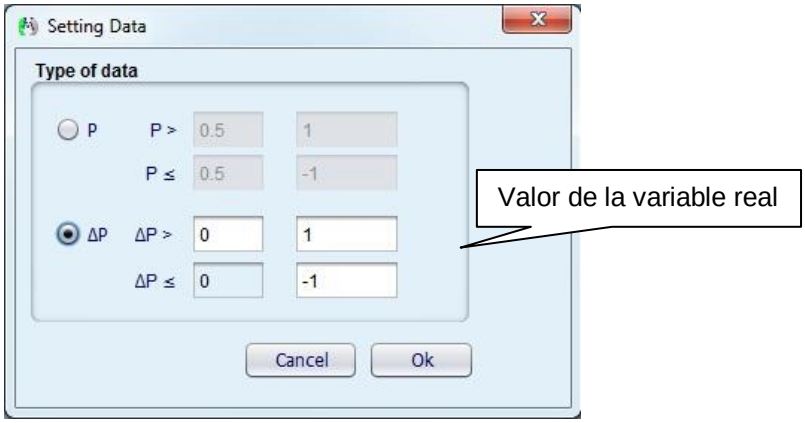

Figura 3.5 Seleccionar tipo de datos.

Al momento de seleccionar el tipo de dato el software le ofrece al usuario una información de error si no coincide el valor de clase seleccionada con los reales cargados.

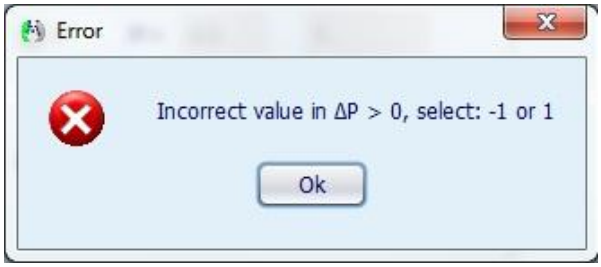

Figura 3.6 Valor de clase incorrecto

3.1.5 Segundo paso: pre-procesamiento ("*Preprocess*")

El segundo paso es un pre-procesamiento de los modelos individuales donde se obtienen estadísticas de cada uno de ellos. Como se muestra en la Figura 3.7 aparece una lista con todos los modelos que contiene la data de entrada.

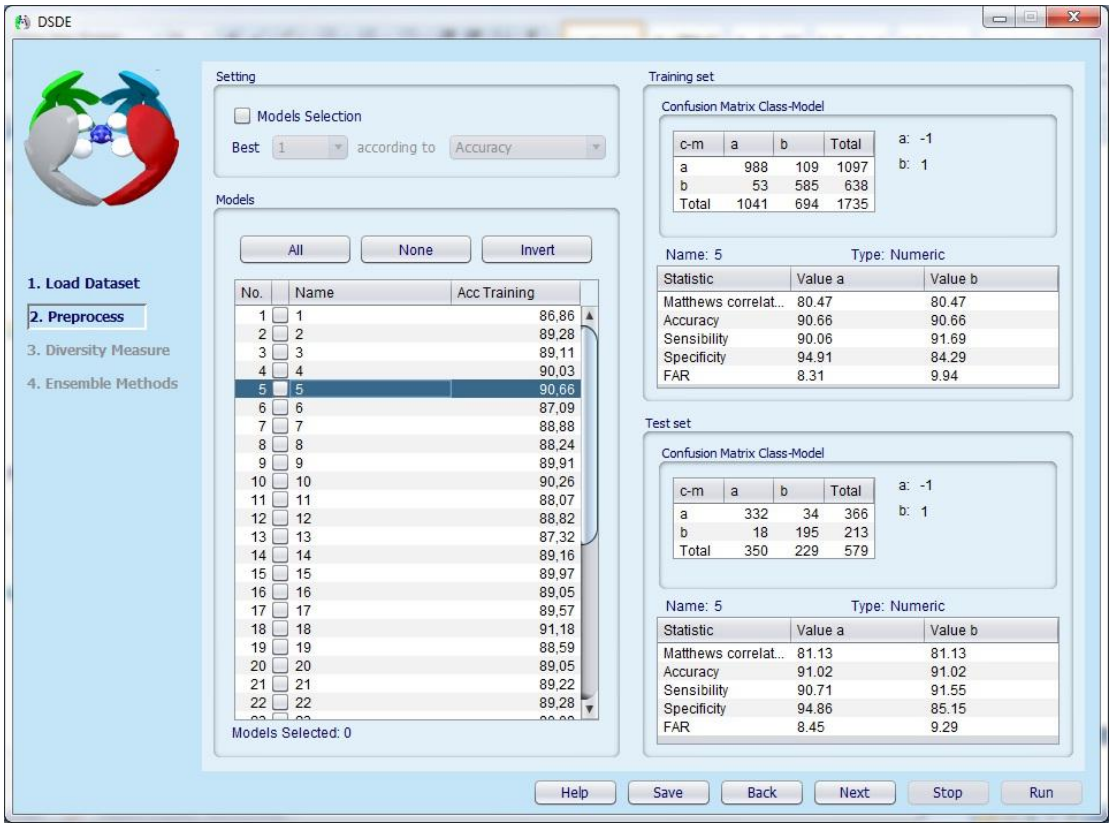

Figura 3.7 Ventana de pre-procesamiento. Vista de los datos.

Para obtener las estadísticas basta con dar click sobre el modelo a analizar, mostrándose los resultados automáticamente en la parte derecha de la ventana. Para pasar al siguiente paso es necesario que el usuario seleccione los modelos con los cuales desea continuar el proceso. De oprimir el botón "*Next*" sin haber seleccionado al menos tres modelos, le muestra al usuario la siguiente información:

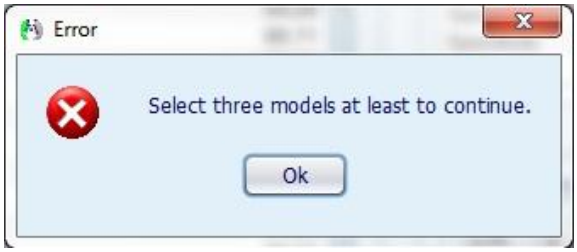

Figura 3.8 Cantidad de modelos a seleccionar como mínimo

La selección de los modelos individuales para pasar al paso de las medidas de diversidad, se puede hacer de las siguientes formas:

1- Marcar manualmente la casilla de selección que aparece al lado de cada modelo (Figura 3.9)

2- Seleccionar todos ("***All"***)

3- Seleccionar algunos modelos e invertir la selección ("***Invert"***)

4- Selección automática con la opción "***Models Selection".*** El usuario debe seleccionar la cantidad de modelos que quiere obtener a partir de un parámetro estadístico. En este caso la selección se hace teniendo en cuenta los mayores valores de Exactitud o Coeficiente de Correlación de Matthews (Figura 3.10)

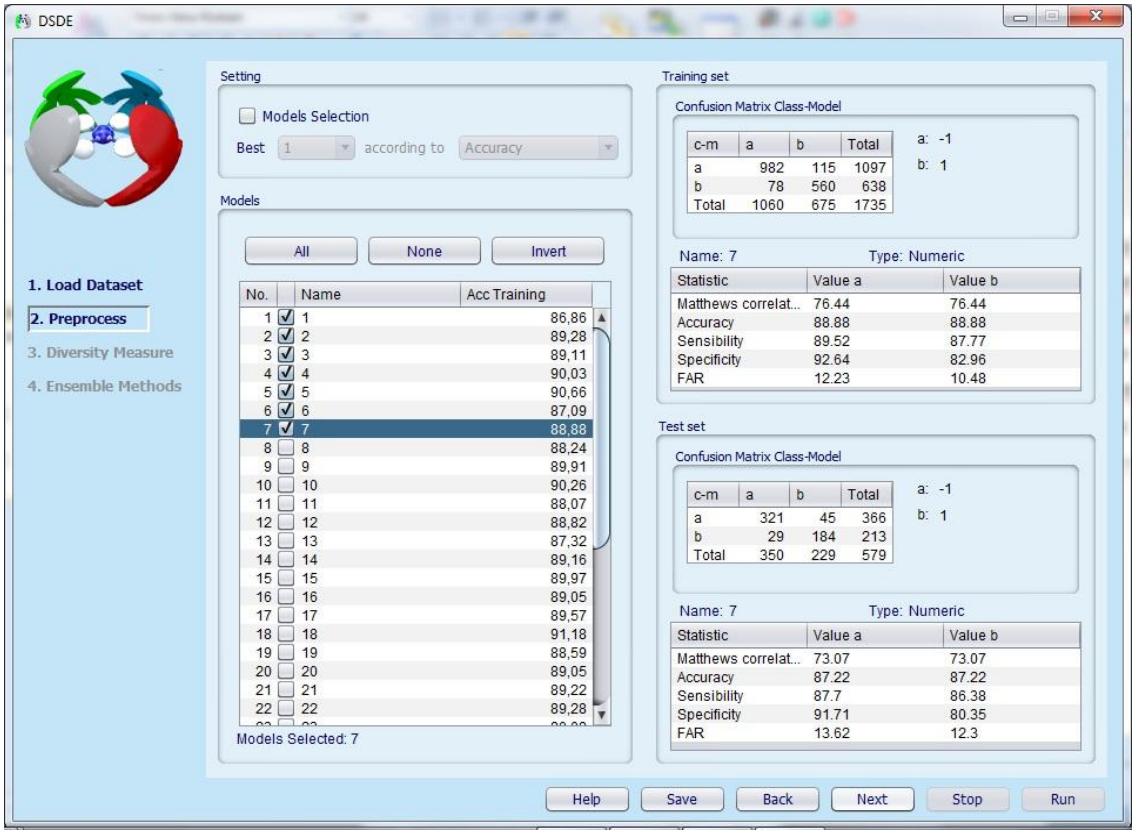

Figura 3.7 Selección de modelos de forma manual.

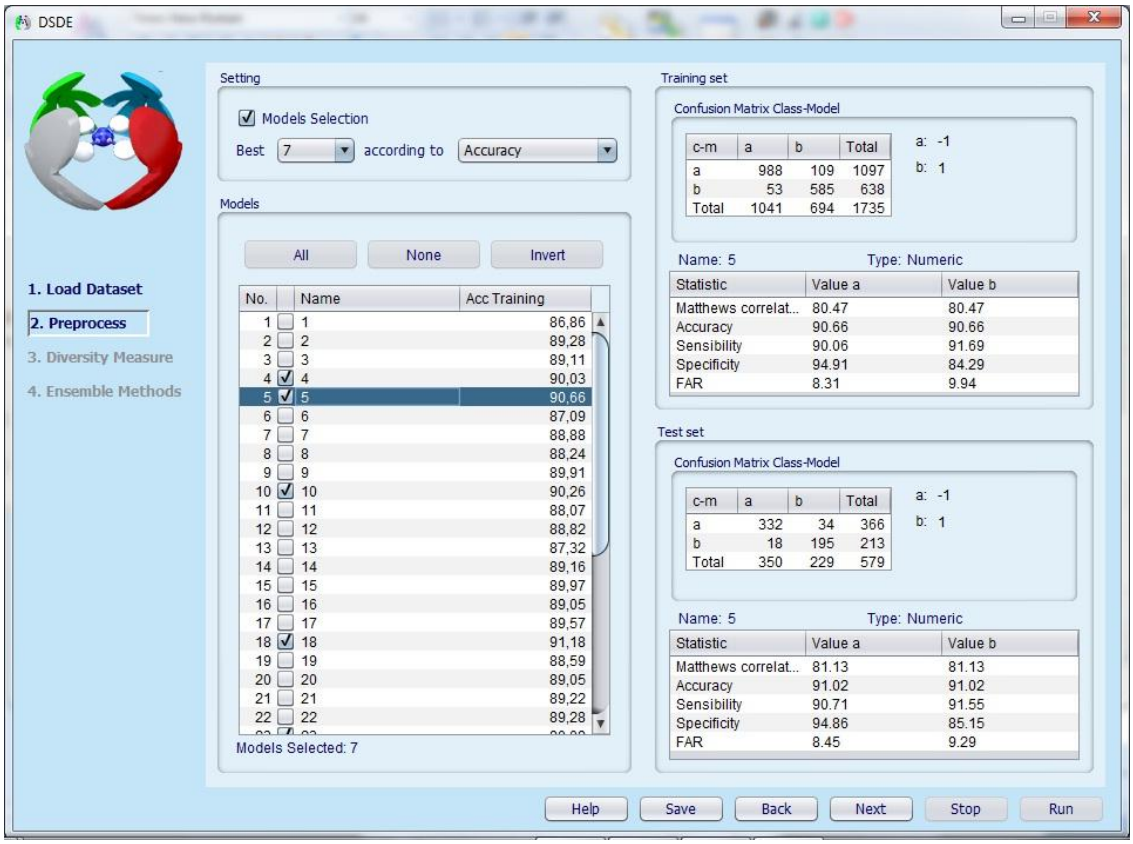

Figura 3.8 Selección de los modelos de forma automática.

En la parte inferior de la pantalla aparecen los botones de ayuda, guardar, anterior o siguiente.

"***Help***" Muestra la ayuda al usuario.

"***Save***" Guarda en fichero .txt todos los resultados estadísticos para todos los modelos individuales.

"***Back***" Regresa al Primer paso: Cargar datos ("***Load Dataset***"). El usuario debe volver a seleccionar el tipo de data para pasar a la ventana actual.

"***Next***" Pasa el siguiente paso Medidas de Diversidad ("***Diversity Measure***").

Los botones "***Stop***" y "***Run***" aparecen desactivados en este paso.

3.1.6 Tercer paso: aplicar medidas de diversidad ("*Diversity Measure*")

La Figura 3.11 muestra la ventana donde se realiza las medidas de diversidad, la cual tiene las siguientes particularidades:

- Solo se puede seleccionar una medida y una combinación cada vez.
- Se debe especificar la cantidad de mejores combinaciones a obtener en los resultados; el proceso incluye todos los modelos que aparecen en la lista.
- Los resultados se muestran por defecto para la Serie de Entrenamiento y opcionalmente se puede seleccionar la casilla "*Test Set*" para mostrar los resultados de la Serie de Predicción.
- El software muestra por defecto el valor 1 para mostrar los resultados de diversidad por pantalla. Si la opción seleccionada por el usuario es mayor que la cantidad posible de combinaciones se mostrará una ventana de "***error***".
- En esta ventana ya se muestran activos los botones "***Stop***" y "***Run***".

  "***Run***" Ejecuta las opciones de medida de diversidad seleccionada y la combinación posible.

  "***Stop***" Detiene el proceso de selección de medidas de diversidad.

  El panel inferior derecho muestras los resultados de las medidas de diversidad. Según estos resultados se irán seleccionando los modelos de forma automática para pasar al siguiente paso: Métodos de ensamble ("***Ensemble Methods***").

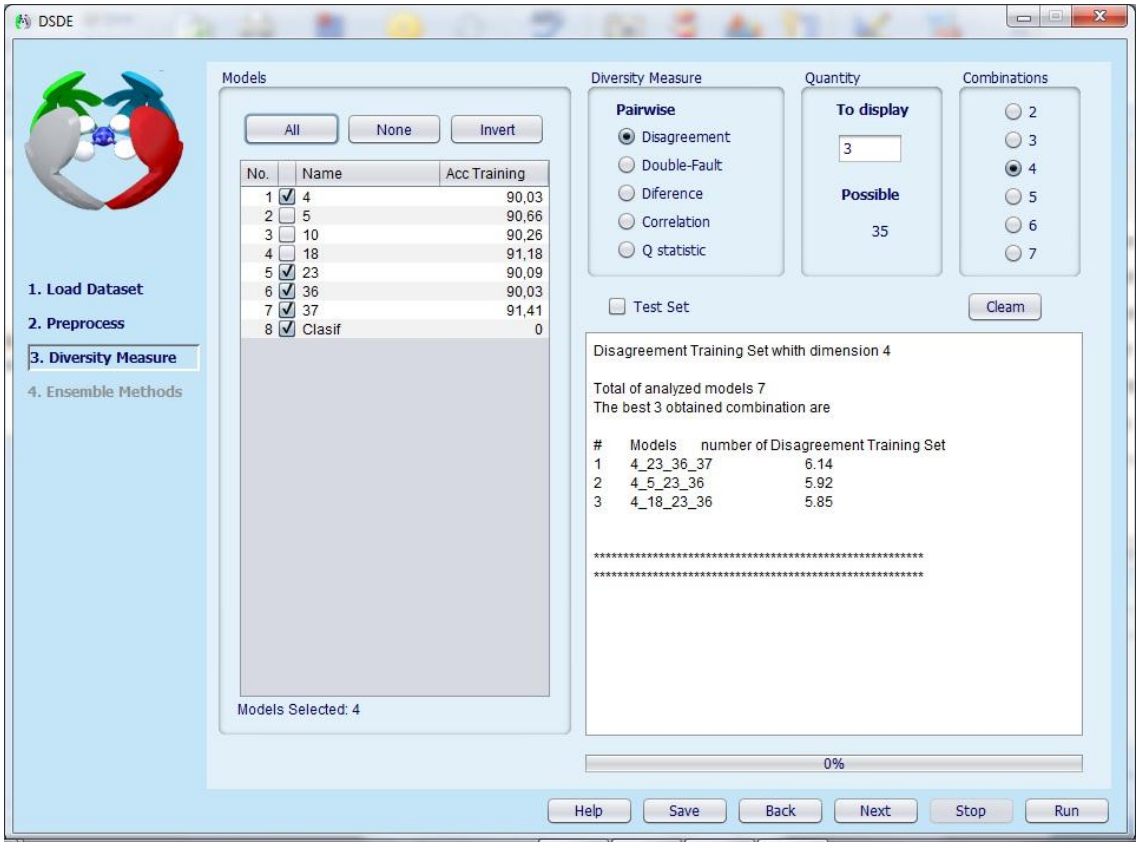

Figura 3.9 Vista de las medidas de diversidad

3.1.7 Cuarto paso: ensamble de los modelos individuales

Para realizar el ensamble de los modelos individuales el usuario puede seleccionar varios métodos y combinaciones. Además, si desea cambiar el valor a partir del cual se define cada clase puede hacerlo dando click en el botón "***Setting***" (Figura 3.12).

El usuario tiene la opción de escoger la forma en que desea visualizar los mejores modelos ensamblados mediante el botón "***Selection***" (Figura 3.13).

Para ejecutar ("***Run***") y detener ("***Stop***") el proceso se hace de la misma forma que en el paso anterior.

Los nuevos modelos ensamblados se agregan a la lista de modelos individuales, mostrando además el valor de la exactitud. En la ventana de la parte inferior derecha se muestra un resumen de los resultados obtenidos (Figura 3.14).

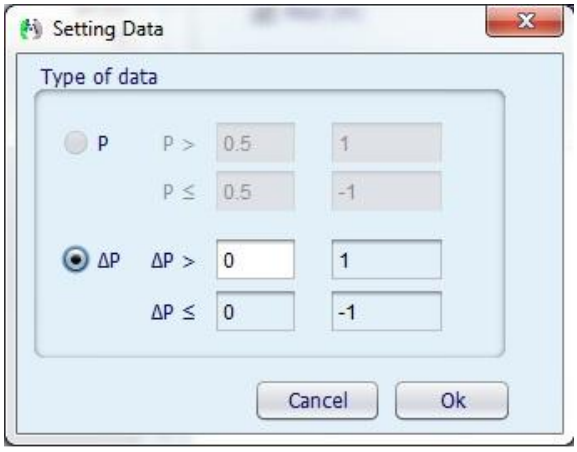

Figura 3.10 Ventana de diálogo: Configuración de los datos.

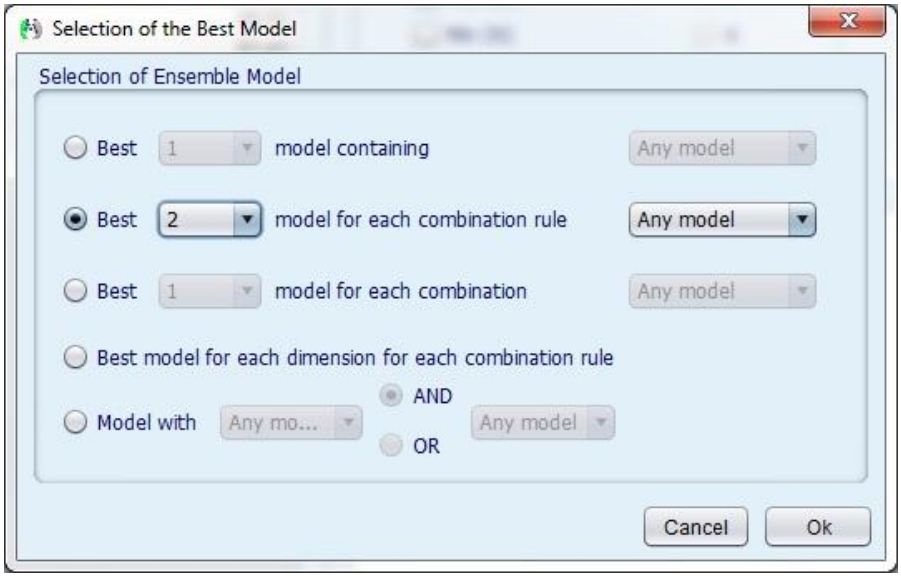

Figura 3.11 Ventana de diálogo: Selección de los mejores modelos.

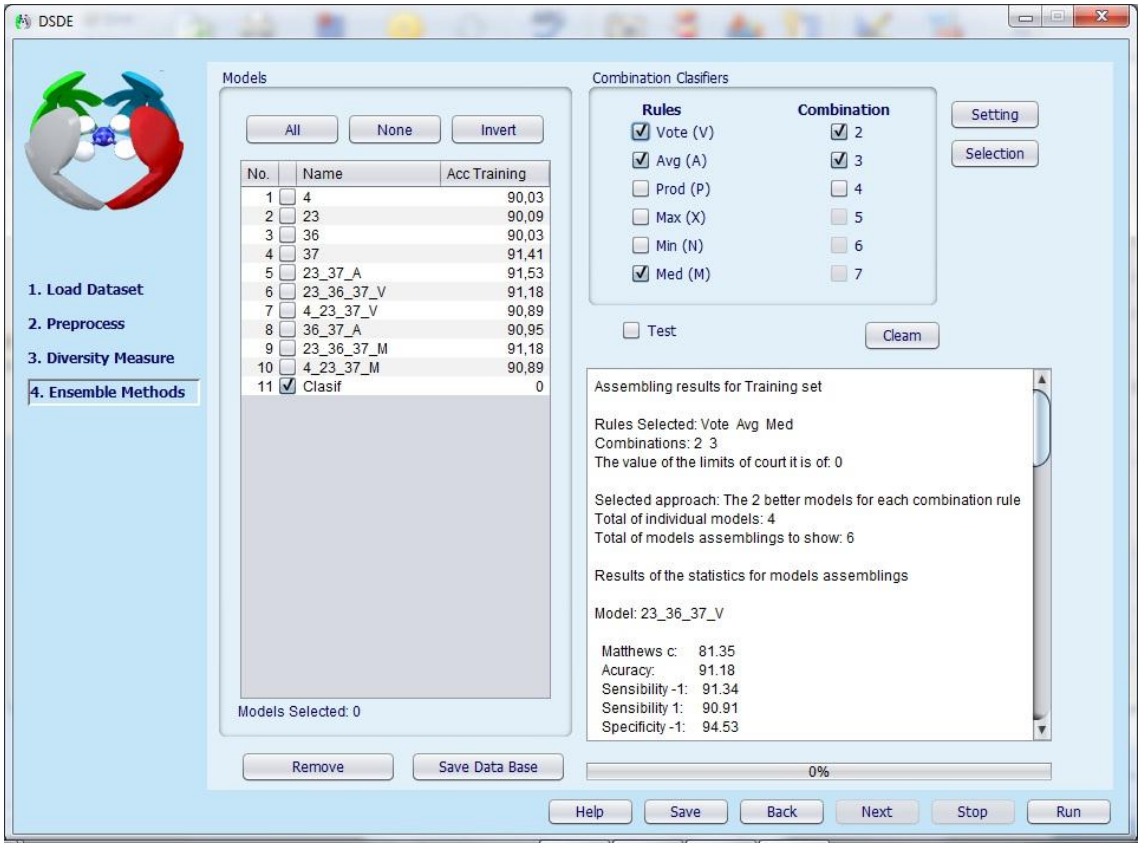

Figura 3.12 Vista del ensamble de modelos individuales.

3.2 Análisis de los resultados obtenidos

Tres bases de datos de modelos *QSAR* (*Quantitive Structure Activity Relationship*) fueron evaluadas para validar el software DSDE. Estas bases de datos fueron obtenidas por investigadores del Centro de Bioactivos Químicos y de la UCLV para identificar compuestos antimaláricos, antileishmaniales y antinflamatorios.

La Malaria y la Leishmaniosis son enfermedades protozoarias de alto impacto social, provocando la muerte de 1-3 millones de personas y de 300-500 millones de afecciones anualmente. En la actualidad existe una urgente necesidad de descubrir nuevas alternativas terapéuticas, dado que los fármacos disponibles tienen problemas de toxicidad y/o resistencia. Universidades e instituciones pueden jugar un papel importante en la búsqueda de nuevas estrategias terapéuticas, y tienen el potencial para crear nuevos paradigmas de descubrimiento de fármacos antiprotozoarios que aumenten la efectividad y eficiencia de los métodos tradicionales de experimentación de "*prueba y error*".

En el proceso de cribado virtual para la selección de compuestos potencialmente antimaláricos, se obtuvieron primero los modelos individuales. Para ello se utilizó una base de datos de 2 314 compuestos que se dividieron en 851 y 1 463 activos e inactivos, respectivamente, utilizando un análisis de *clúster* de *k*-NNCA (*k*-vecinos más próximos), implementado en el paquete estadístico *STATISTICA 6.0*. Mediante un análisis de *clúster* del tipo *k*-MCA (de *k*-Medias) (Johnson and Wichern 1988; Mc Farland and Gans 1995; Xu and Hagler 2002), quedó constituida la serie de entrenamiento (SE) con 638/1097 y serie de predicción (SP) con 213/366 activos/inactivos, respectivamente, quedando asegurada la representatividad de elementos del mismo dominio en los dos subconjuntos obtenidos. Todos los compuestos fueron calculados usando el descriptor molecular *TOMOCOMD CARDD (TOpological Molecular COMputer Design - Computer Aided Rational Drug Design*). En total fueron obtenidos 37 modelos por Análisis Discriminante Lineal usando los índices lineales, bilineales y cuadráticos y basados en las relaciones de átomos.

Igual procedimiento se siguió para la confección de la base de datos para identificar compuestos antileishmaniales. Se partió de una base datos constituida por 196 173 productos probados *in vitro* contra promastigotes de *Leishmania major (SHARLOW, CLOSE et al. 2009)*. De ellos se seleccionaron 10 000 al azar, siendo la mitad activos a 10µM y la otra mitad inactivos. A partir de las estructuras químicas se calcularon los descriptores moleculares implementados en el programa DRAGON y se buscaron modelos de clasificación por análisis discriminante lineal empleando el paquete estadístico STATISTICA. Se obtuvieron 47 modelos, de ellos 27 con las variables pertenecientes a las familias de descriptores 0-2D, 19 modelos con variables 3D y uno que incluyó variables 0-3D.

La tercera base de datos evaluada corresponde a 44 modelos QSAR para identificar posible actividad antiinflamatoria. Para la confección de los modelos se partió de una base de datos de 1 213 compuestos, los cuáles se dividieron en 587 reportados con actividad antiinflamatoria y 626 inactivos. La serie de entrenamiento quedó constituida con 443/476 activos/inactivos y la serie de predicción con 144/150 activos/inactivos respectivamente.

Las tres bases de datos de modelos se obtuvieron utilizando el método de "pasos hacia delante" (*forward stepwise*) y de "mejor subconjunto" (*best subset*) para la selección de las variables, a través de un análisis discriminante lineal (*ADL*).

Diferentes parámetros estadísticos se evaluaron en el paso 2, *Preprocess*, para comprobar la calidad y robustez de todos los modelos obtenidos: exactitud total (*Q*), coeficiente de correlación de Matthews (*C*), sensibilidad (*Sens*), especificidad (*Spec*) y razón de falsa alarma (*FAR)*.

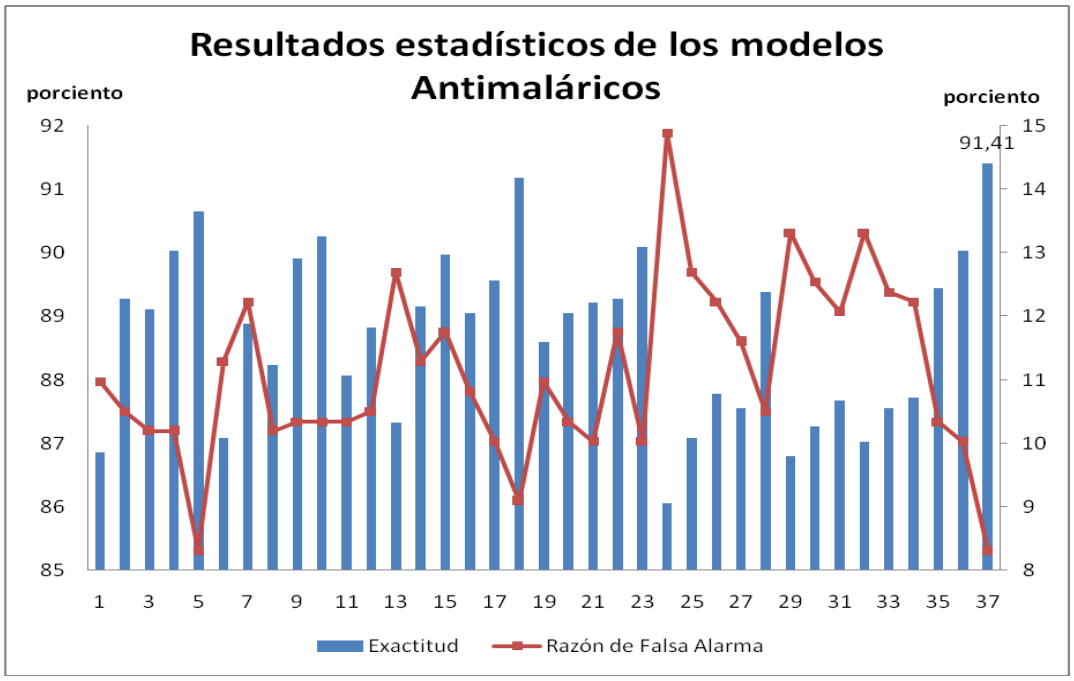

Figura 3. 13 Resultados del pre-procesamiento de la actividad antimalárica

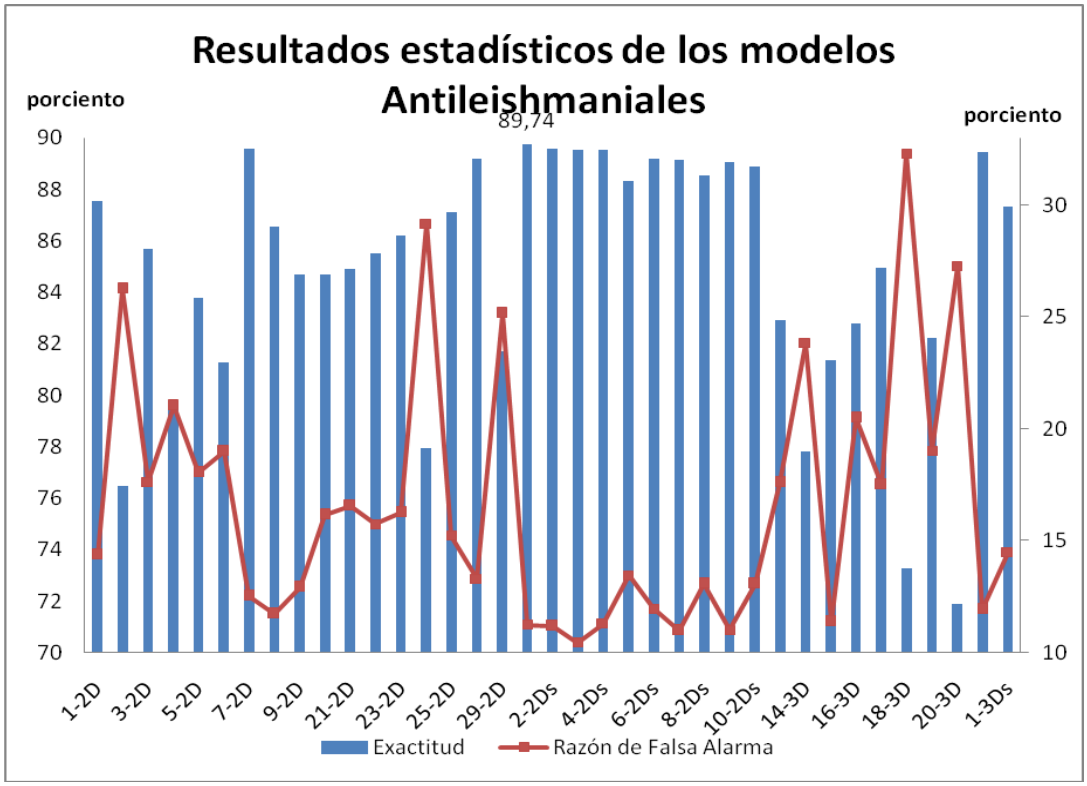

Figura 3. 14 Resultados del pre-procesamiento de la actividad antileishmanial

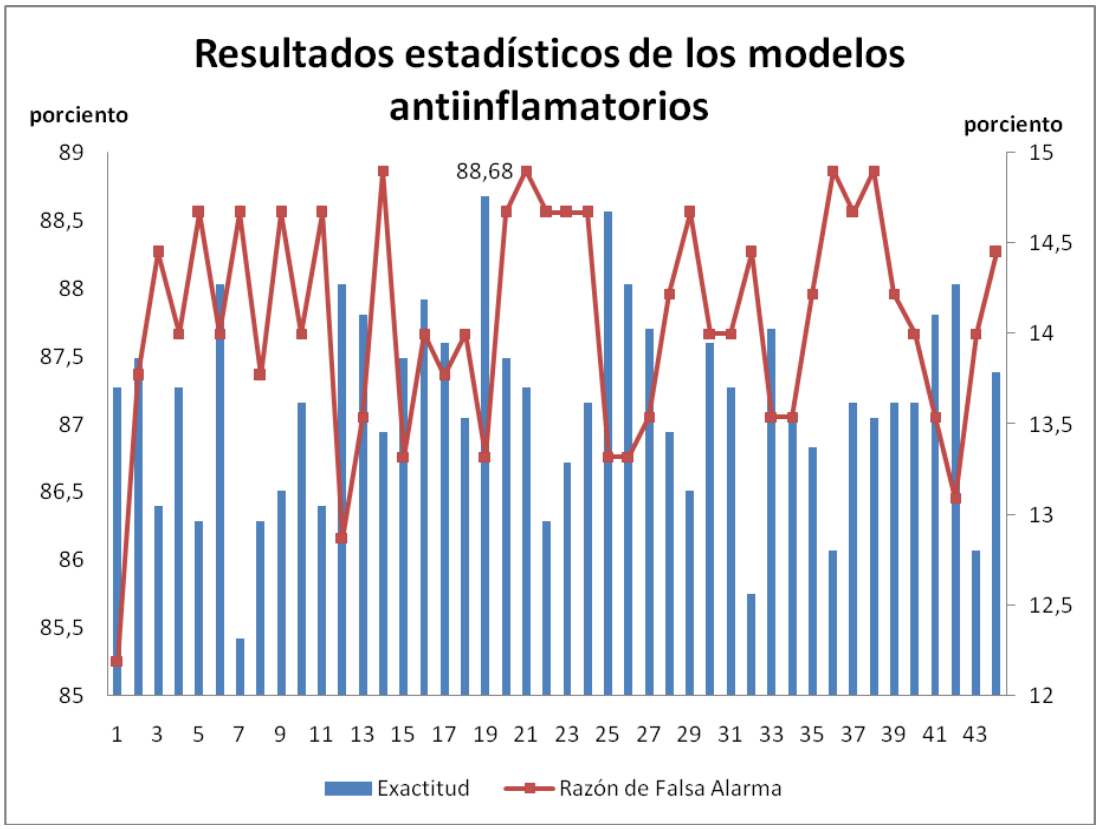

Figura 3. 15 Resultados del pre-procesamiento de la actividad antiinflamatoria

Para "cuantificar" las correlaciones entre todos los clasificadores individuales, fueron seleccionadas todas las medidas de diversidad pareadas implementadas en el paso 3 del software, *Diversity Measure*, con dimensiones 2, 3 y 4. El software selecciona automáticamente los modelos que estén en la primera posición.

Una vez obtenidos los mejores modelos se procede al análisis en el paso 4, *Ensemble Methods*, con todas las reglas y todas las combinaciones.

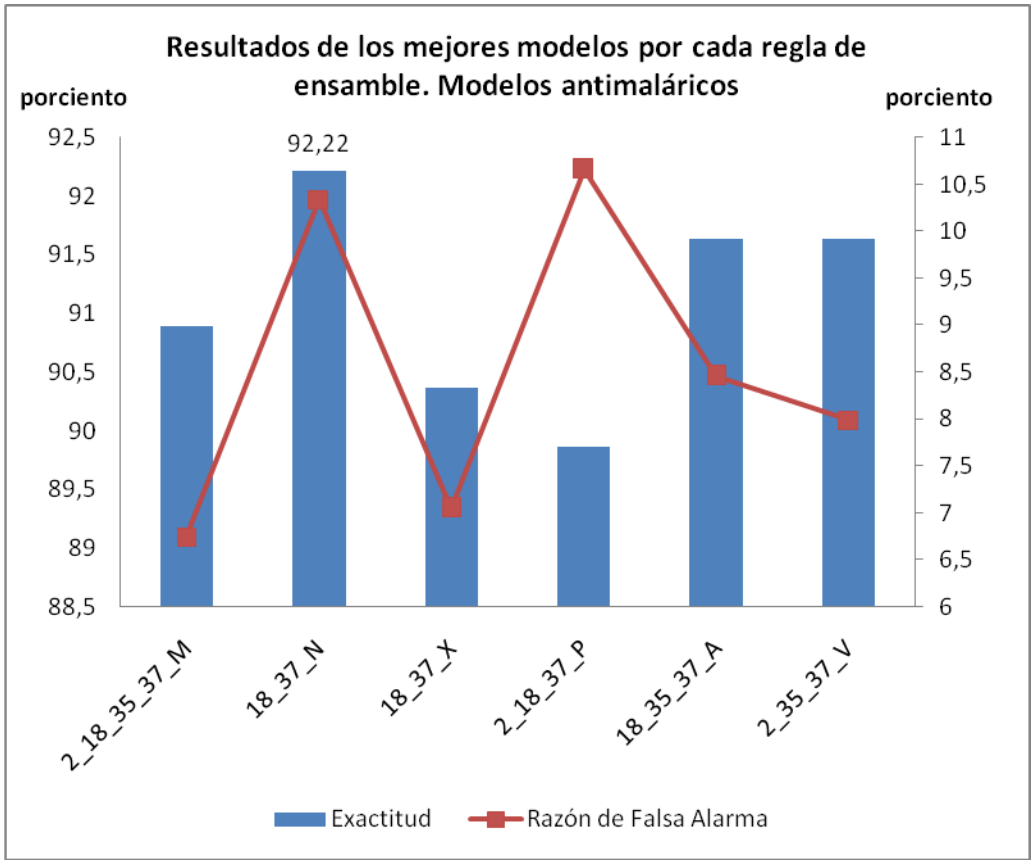

Figura 3. 16 Resultados del ensamble de la actividad antimalárica.

La combinación de los modelos 18 y 37 usando la regla Mínimo (N) obtuvo mejor exactitud (92,22 %) en comparación con el mejor modelo individual 37 (91,41 %).

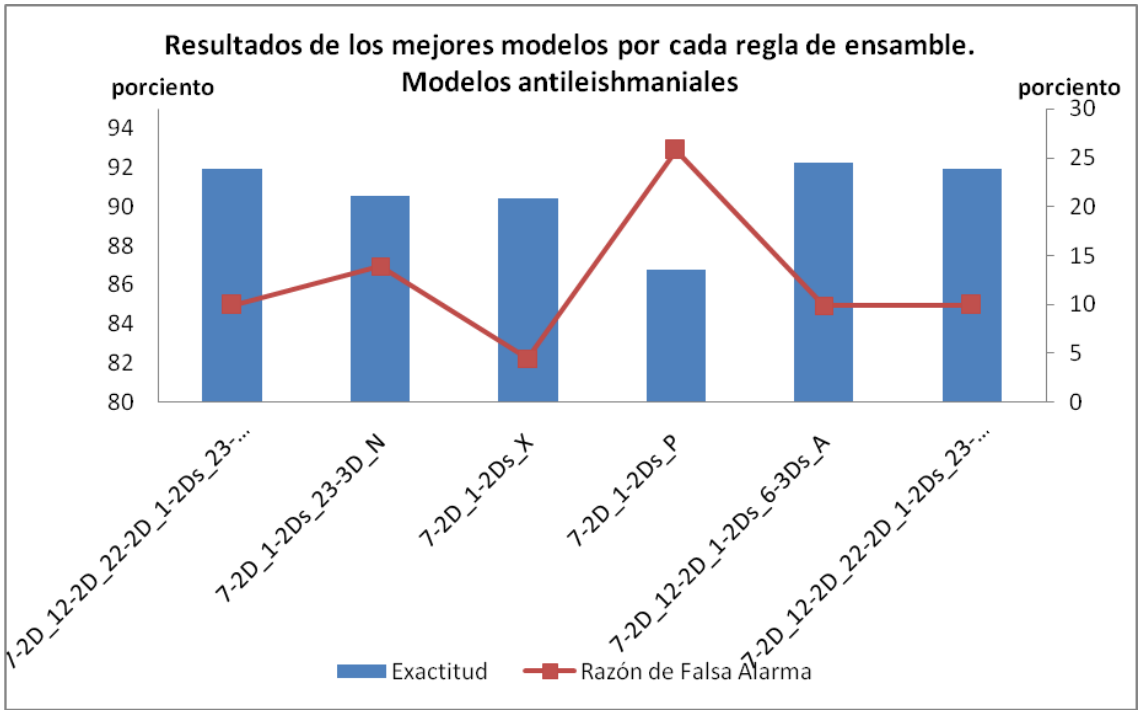

Figura 3. 17 Resultados del ensamble de la actividad antileishmanial.

La combinación de los modelos 7-2D, 12-2D, 1-2Ds y 6-3Ds usando la regla Average (A) obtuvo mejor exactitud (92,26 %) en comparación con el mejor modelo individual 18 (89,74 %).

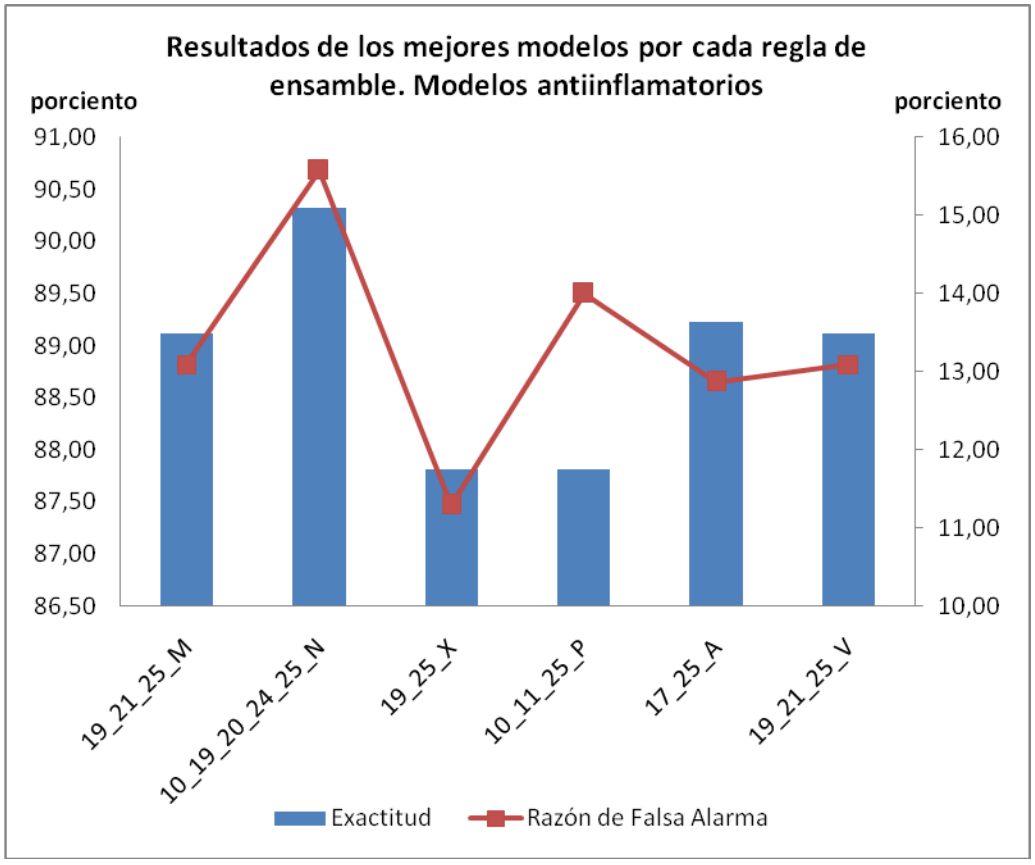

Figura 3. 18 Resultados del ensamble de la actividad antiinflamatoria.

La combinación de los modelos 10, 19, 20, 24 y 25 usando la regla Mínimo (N) obtuvo mejor exactitud (90,32 %) en comparación con el mejor modelo individual 19 (88,68 %).

3.3 Consideraciones finales del capítulo

En el presente capítulo se expusieron los aspectos fundamentales para el uso del software DSDE 1.0 especificando requerimientos, funcionalidades y forma de trabajar con el mismo. Se mostraron tres aplicaciones con datos reales, en todas se obtuvieron buenos resultados con el software, demostrando que es posible encontrar multiclasificadores con mejores resultados de exactitud que los modelos individuales.

CONCLUSIONES

CONCLUSIONES

De los resultados presentados se derivan las siguientes conclusiones:

1. Se seleccionaron e implementaron en el software DSDE diferentes parámetros estadísticos para corroborar la calidad y robustez de los clasificadores individuales. Además, se incluyeron las medidas de diversidad pareadas reportadas en la literatura
2. Se implementaron seis estrategias de combinación empleando el enfoque de fusión, permitiendo así obtener modelos ensamblados.
3. Se diseñó e implementó una interfaz gráfica que permite de forma automática y amigable, analizar estadísticas, la diversidad y la selección de los mejores modelos ensamblados. El sistema se implementó sobre una plataforma de software libre.
4. Para la validación de la herramienta se utilizaron tres bases de datos reales a través de las cuales se pudo comprobar que es posible obtener los clasificadores que mejor combinan y por tanto mejoren su predicción.

RECOMENDACIONES

RECOMENDACIONES

En la continuación del software presentado es importante se consideren las siguientes recomendaciones:

1. Implementar medidas de diversidad no pareadas reportadas en la literatura.
2. Realizar pruebas de comparación múltiple de tipo post hoc.
3. Desarrollar e implementar nuevos métodos de combinación.
4. Lograr desarrollar cribados virtuales de grandes bases de datos a partir de los modelos ensamblados.
5. Realizar análisis de clúster para poder identificar y reducir los compuestos seleccionas en el cribado virtual.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

Baldi, P., S. Brunak, et al. (2000). "Assessing the accuracy of prediction algorithms for classification: an overview." Bioinformatics Review **16**: 412-424.

Bello García, M. (2012). Un método de aproximación de funciones basado en el enfoque de los prototipos más cercanos utilizando relaciones de similaridad. Tesis para optar por el título de licenciatura en computación. Santa Clara, Universidad Central "Marta Abreu" de las Villas.

Bezdek, J. C., M. R. Pal, et al. (1999). "Fuzzy models and algorithms for pattern recognition and image processing." Pattern Recognition and Inage Processing.

Bonet Cruz, I. (2008). Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Centro de Estudios Informáticos. Santa Clara, Universidad Central Marta Abreu de las Villas.

Bulacio, P. E. (2006). Sistema de Clasificadores Heterogéneos para toma de decisión conjunta basada en agregación de información mediante Integrales Borrosas. Departamento de ingeniería de Sistemas Telemáticos. Tesis Doctoral. Madrid, Universidad Politécnica de Madrid. Ingeniera Electrónica**:** 148.

Cho, S. B. and J. Kim (1995). "Combining multiple neural networks by fuzzy integral and robust classification." IEEE Transactions on Systems, Man, and Cybernetics **25**: 380-384.

Dietterich, T. G. 2000. Ensemble methods in machine learning. *Multiple Classifier Systems.* Berlin: Springer-Verlag Berlin. 1-15

Drucker, H., C. Cortes, et al. (1994). "Boosting and other ensembles methods." Neural Computation **6**: 1289-1301.

Fisher, D. H. (1987). *Knowledge Acquisition via Incremental Conceptual Clustering*. Machine Learning. 2:139-172, reprinted in Shavlik & Dietterich

(eds.), Readings in Machine Learning, section 3.2.1.

Fowler, M. and K. Scott (1997). "UML Distilled." Massachusetts, E.U.A Addison Wesley Longman.

Giacinto, G. and F. Roli (2001). "Desing of effective neural network ensembles for image classification purposes." Image and Vision Computing **19(9-10)**: 699-707.

Giacinto, G., F. Roli , et al. (2000). "Selection of Classifiers based on Multiple Classifier Behaviour." 1876: 87-92.

Gonell, Sara. S.-S. (2010). "Métodos de reducción de la carga computacional de clasificadores multiclase basados en máquinas de vectores soporte." Tesis para optar por el título ingeniero en telecomunicaciones. Universidad Carlos III. Leganés, Madrid, España. 43-46.

Gonzalez-Diaz, H., M. A. Dea-Ayuela, et al. (2008). "HP-Lattice QSAR for dynein proteins: Experimental proteomics (2D-electrophoresis, mass spectrometry) and theoretic study of a Leishmania infantum sequence." Bioorganic & Medicinal Chemistry Letters **16**(16): 7770-7776.

Hansen, L. and P. Salomon (1990). "“Neural Networks Ensembles”." IEEE Trans. Pattern Analysis and Machine Intelligence **12**: 993-1001.

Jacobs, R. (1996). "Methods for combining experts’ probability assessments." Neural Computation **7**: 867–888.

Jacobson, I., G. Booch, et al. (2000). "El Proceso Unificado de Desarrollo de Software."

Jain, A. K. and R. P. W. Y. M. J. C. Duin (2000). "“Statistical pattern recognition: a review”." IEEE Trans. Pattern Analysis and Machine Intelligence **22**(1): 4-37.

Johnson, R. A. and D. W. Wichern (1988). Applied Multivariate Statistical Analysis. Englewood Cliffs, NJ, Prentice-Hall.

Kam, H. T. (1998). "The random subspace method for constructing decision forests." IEEE Trans. Pattern Anal. Mach. Intell. **20**(8): 832–844.

Kamel, M. S. and N. M. Wanas (2003). "Data dependence in combining classifiers,." LECT NOTES COMPUT SC 2709: 1-14.

Kittler , J. and F. M. Alkoot (2003). "Sum versus vote fusion in multiple classifier systems." IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(1): 110-115.

Kittler, J., M. Hatef, et al. (1996). "Combining classifiers." IEEE Trans. **2**(B): 897-901.

Kittler, J., M. Hatef, et al. (2002). "On combining classifiers." Pattern Analysis and Machine Intelligence. **20**(3): 226–239.

Kittler, J., M. Hatef, et al. (1998). "On combining classifiers." IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3): 226-239.

Kohavi, R. and D. H. Wolpert (1996). Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, Machine Learning: Proceedings of the Thirteenth International Conference, pages 275–283. Morgan Kaufmann.

Krogh, A. and J. Vedelsby (1995). Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems, volume 7, pages 231–238. Krzanowski, W. and D. Partridge (1995). Software Diversity: Practical Statistics for its Measurement and Exploitation. Department of Computer Science. Prince of Wales Road, Exeter, University of Exeter.

Kuncheva, L. and C. A. Shipp (2002). "Relationships between combination methods and measures of diversity in combining classifiers." Information Fusion **3**(2): 135–148.

Kuncheva, L. I., Jain, L.C. (2000). "Designing classifier fusion systems by genetic algorithms." IEEE Transactions on Evolutionary Computation. **4**(4): 327-336.

Kuncheva, L. I. and C. A. Shipp (2002). "Relationships between combination methods and measures of duversity in combining classifiers." Information Fusion **3(2)**: 135-148.

Kuncheva, L. I. and C. J. Whitaker (2002). "Measure of diversity in classifier ensemble and their relationship with the ensemble accuaracy." Machine Learning **51(2)**: 181-207.

Kuncheva, L. I. and C. J. Whitaker (2002). "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy." Machine Learning **51**(2): 181–207.

L. Xu, A., Krzyzak, et al. (1992). "Methods of combining multiple classifiers and their applications to hand-written character recognition." IEEE trans. on Systems, Man and Cybernetics **22**(3): 418–435.

Lam, L. (2000). Classifier combinations: implementations and theoretical issues. Multiple Classifier Systems. J. Kittler and F. Roli. Cagliari, Italia, Springer. Lecture Notes in Computer Science: 78-86.

Last, M., H. Bunke, et al. (2002). "A feature-based serial approach to classifier combination." Pattern Analysis & Applications **5**: 385-398.

Lynch, R. S. and P. K. Willet (2003). "Classifier fusion results using various open literature data sets." IEEE International Conference on Systems, Man and Cybernetics **1**: 723-728.

Maclin, R. and J. Shavlik (1995). Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Montreal, Canada.

MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press.

Marrero-Ponce, Y., Y. Machado-Tugores, et al. (2005). "A computer-based approach to the rational discovery of new trichomonacidal drugs by atom-type linear indices." Curr Drug Discov Technol **2**(4): 245-65.

Marrero-Ponce, Y., A. Meneses-Marcel, et al. (2006). "Predicting antitrichomonal activity: a computational screening using atom-based bilinear indices and experimental proofs." Bioorg Med Chem **14**(19): 6502-24.

Marrero-Ponce, Y., A. Meneses-Marcel, et al. (2008). "Bond-based linear indices in QSAR: computational discovery of novel anti-trichomonal compounds." J Comput Aided Mol Des **22**(8): 523-40.

Marrero-Ponce, Y., O. M. Rivera-Borroto, et al. (2009). "Discovery of Novel Trichomonacidals Using LDA-Driven QSAR Models and Bond-Based Bilinear Indices as Molecular Descriptors." Qsar & Combinatorial Science **28**(1): 9-26.

Mc Farland, J. W. and D. J. Gans (1995). Cluster Significance Analysis. Chemometric Methods in Molecular Design. H. Waterbeemd. Weinheim, Ger, VCH Publishers**:** 295–307.

Meneses-Marcel, A., Y. Marrero-Ponce, et al. (2005). "A linear discrimination analysis based virtual screening of trichomonacidal lead-like compounds: Outcomes of in silico studies supported by experimental results." Bioorganic & Medicinal Chemistry Letters **15**(17): 3838-3843.

Meneses-Marcel, A., O. M. Rivera-Borroto, et al. (2008). "New antitrichomonal drug-like chemicals selected by bond (edge)-based TOMOCOMD-CARDD descriptors." J Biomol Screen **13**(8): 785-94.

Montero-Torres, A., R. N. Garcia-Sanchez, et al. (2006). "Non-stochastic quadratic fingerprints and LDA-based QSAR models in hit and lead generation through virtual screening: theoretical and experimental assessment of a promising method for the discovery of new antimalarial compounds." Eur J Med Chem **41**(4): 483-93.

Montero-Torres, A., M. C. Vega, et al. (2005). "A novel non-stochastic quadratic fingerprints-based approach for the 'in silico' discovery of new antitrypanosomal compounds." Bioorg Med Chem **13**(22): 6264-75.

Rahman, A. F. R. and M. C. Fairhurst (2003). "Multiple classifier decision combination strategies for character recognition: A review." International Journal on Document Analysis and Recognition **5**: 166-194.

Rumbaugh, J. and G. Booch (2000). "El Lenguaje Unificado de Modelado."

Ruta, D. and B. Gabrys (2001). "Analysis of the Correlation Between Majority Voting Error and the Diversity

Measures in Multiple Classifier Systems, in Soft Computing and Intelligent Systems for Industry: Proceedings

and Scientific Program:." Fourth International ICSC Symposium 20012001 ICSC-NAISO Academic Press:

Paisley, Scotland. p. 50.

Segrera, F. S. and G. M. N. Moreno (2006). Multiclasificadores: Métodos y Arquitecturas. España, Universidad de Salamanca**:** 47 (1-15).

Sharkey, A. and N. Sharkey (1995). "How to improve the reliability of artificial neural networks.".

Sharlow, E. R., D. Close, et al. (2009). Identification of potent chemotypes targeting Leishmania major using a high-throughput, low-stringency, computationally enhanced, small molecule screen. .

Skalak, D. B. (1996). The sources of increased accuracy for two proposed boosting algorithms. In P. Chan, editor, Working Notes of the AAAI Workshop on Integrating Multiple Learned Models, pages 120–125,

Smyth, P. (1995). "Bounds on the mean classification error rate of multiple expert.".

Toh, K. A. (2004). "Yau, W.Y.: Combination of hyperbolic functions for multimodal biometrics data fusion." IEEE Transactions on Systems, Man and Cybernetics **34**(2): 1196-1209.

Valentini, G. and F. Masulli (2002). Ensembles of learning machines. Series Lecture Notes in Computer Sciences. M. Marinaro and R. Tagliaferri. Heidelberg, Alemania, Neural Nets WIRN Vietri.

Vega, M. C., A. Montero-Torres, et al. (2006). "New ligand-based approach for the discovery of antitrypanosomal compounds." Bioorg Med Chem Lett **16**(7): 1898-904.

Webb, G. I. (2000). "Multiboosting: a technique for combining Boosting and Wagging." 159-196.

Xu, J. and A. Hagler (2002). "Chemoinformatics and Drug Discovery." Molecules **7**: 566-700.

Yan, W. and K. Goebel (2004). "Designing Classifier Ensembles with Constrained Performance Requirements, Proceedings of SPIE Defense & Security Symposium, Multisensor Multisource Information Fusion: Architectures, Algorithms, and Applications."

Yule, G. U. (1990). "On the association of attributes in statistics." Philosophical Transactions of the Royal Society of London. **194**(A): 257–319.