Universidad Central "Marta Abreu" de las Villas

Facultad de Matemática-Física-Computación

Lic. Ciencias de la Computación

Trabajo de Diploma

Título:

Herramienta Computacional DASDE® para aplicaciones quimio-bioinformáticas usando Algoritmos Genéticos en la construcción de multiclasificadores

Autora:

Claudia Peña Díaz

Tutores:

Lic.: Maricel Meneses Gómez

MSc.: Leidys Cabrera Hernández

Santa Clara, Cuba

2016

DICTAMEN

Claudia Peña Díaz, hago constar que el trabajo titulado “Herramienta Computacional DASDE® para aplicaciones quimio-bioinformáticas usando Algoritmos Genéticos en la construcción de multiclasificadores.” 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

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

"Piensa como piensan los sabios, mas habla como habla la gente sencilla".

Aristóteles

DEDICATORIA

A mis padres por haberme dado la vida.

A mi novio Osniel por apoyarme en cada momento.

A mis hermanas por apoyarme y quererme.

AGRADECIMIENTOS

A nuestros tutores Maricel, Leidys y Alfredo por el tiempo dedicado y por sus valiosas ideas

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

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

A mis hermanas, por apoyarme en todo.

A mi amiga Anabel y su familia por ser incondicional conmigo y darme su cariño.

A mis amigas y amigos en general en especial a Ariamna, Leonardo, Rafael, Carlos Rafael, Radner.

A mis compañeros de aula, por todos los recuerdos inolvidables.

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

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

RESUMEN

El descubrimiento de un nuevo medicamento y desarrollo posterior del mismo, son dos fases que comprenden alrededor de 12 años de investigación en países desarrollados, con un costo promedio de 231 millones de dólares por cada nuevo medicamento que sale al mercado. El desarrollo de la ciencia computacional, ha permitido la búsqueda racional de nuevas alternativas terapéuticas, siendo los métodos *in silico* una de las técnicas que mejoran potencialmente el descubrimiento y posterior desarrollo de fármacos. La utilización de multiclasificadores junto con algoritmos de búsqueda heurística son estrategias, que ayudan en el pronóstico del análisis y a disminuir su tiempo para obtener buenos resultados en la búsqueda y posterior desarrollo de compuestos activos frente a enfermedades prioritarias a nivel mundial. El software DASDE v1.0 desarrollado en el año 2013 para investigadores de perfil químico de nuestra universidad tiene varias funcionalidades, dentro de ellas permite la selección de modelos diversos para obtener posteriormente, combinaciones de los mismos donde se mejoren los resultados alcanzados individualmente, estas funcionalidades se realizan de manera exhaustiva lo que conlleva a grandes demoras en las ejecuciones puesto que las aplicaciones en el campo de la quimio-bioinformática siempre cuentan con un gran número de clasificadores y por lo tanto la cantidad de combinaciones que se pueden generar es considerablemente grande. En este trabajo se propone la incorporación al software de la meta heurística Algoritmos Genéticos para resolver el problema combinatorio que se presenta y minimizar el tiempo de búsqueda de las soluciones, garantizándole al usuario obtener combinaciones que sean igualmente buenas en calidad. Finalmente se presenta el análisis con dos bases de datos reales en el campo de la quimio-bioinformática.

ABSTRACT

The discovery of a new drug and subsequent development are two phases comprising about 12 years of research in developed countries, with an average cost of $ 231 million per new drug hitting the market. The development of computational science, has allowed the rational search for new therapeutic alternatives, where *in silico* methods are one of the techniques that potentially improve the discovery and development of drugs. The use of multi classifiers with heuristic search algorithms are strategies that help in forecasting the analysis and decrease time to get good results in the search and subsequent development of active compounds against priority diseases worldwide. DASDE v1.0 software, developed in 2013 for researchers with a chemical profile of our university, has several features, among them it´s allowed the selection of diverse models for later combinations where the results achieved improve individual ones. These features are performed exhaustively which leads to long delays in executions since the applications in the field of chemo-bio-informatics always have a large number of classifiers and therefore the number of combinations that can be generated is considerably large. In this paper the incorporation of meta heuristic genetic algorithms to the software is proposed to solve the combinatorial problem that occurs and to minimize search time of solutions, guaranteeing the user to obtain combinations that are equally good in quality. Finally, the analysis with two real databases in the field of chemo-bio-informatics is presented.

TABLA DE CONTENIDO.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Las enfermedades parasitarias en la actualidad son un azote tanto para los países subdesarrollados como los que están en vías de desarrollo. El descubrimiento de un nuevo medicamento y desarrollo posterior del mismo, son dos fases, que condicionan lograr un nuevo producto que sea útil en la terapéutica. La primera se encarga de asegurar que el compuesto tiene un deseable perfil de actividad, esta fase se estima que dure aproximadamente 43 meses (4 años) como promedio, en aquellos países y transnacionales farmacéuticas con una amplia infraestructura investigativa. En países con un nivel menor de desarrollo esta fase alcanza aproximadamente unos 5-6 años. La segunda fase comprende la de los estudios clínicos y la del registro farmacéutico, y se estima que duren entre 69 meses y 31 meses respectivamente. Todo este largo proceso, desde su obtención hasta su registro comprende un total de 12 años de investigación, con un costo promedio de 231 millones de dólares por cada nuevo medicamento que salga al mercado.(Escalona et al. 2005)

El desarrollo de la ciencia computacional, ha permitido la búsqueda racional de nuevas alternativas terapéuticas, siendo los métodos *in silico* una de las técnicas que mejoran potencialmente el descubrimiento y posterior desarrollo de fármacos. Una de las líneas de investigación en la búsqueda de nuevos compuestos líderes, utilizando grandes bases de datos internacionales, es la construcción de modelos de clasificación, requiriéndose encontrar aquellos modelos que muestren una mayor precisión en la identificación, por lo que se requiere la selección de aquellos que mejoren su poder predictivo.

La utilización de multiclasificadores es una estrategia, que puede aumentar dicha precisión de análisis y obtener buenos resultados en la búsqueda de estas decisiones.

En los últimos años, investigadores del Centro de Bioactivos Químicos (CBQ) y de la Facultad de Farmacia (FM) de la Universidad Central “Marta Abreu” de las Villas (UCLV), han desarrollado varios modelos de predicción usando el descriptor molecular ***TOMOCOMD CARDD*** (Marrero-Ponce et al. 2005; Meneses-Marcel et al. 2005; Marrero-Ponce et al. 2006; Marrero-Ponce et al. 2008; Meneses-Marcel et al. 2008; Marrero-Ponce et al. 2009) y análisis discriminante lineal para identificar nuevos compuestos activos frente a distintas enfermedades parasitológicas.

Para dar continuidad al descubrimiento de nuevos compuestos mediante el uso de modelos QSAR (*Quantitative Structure Activity Relationship*), donde se utilizan grandes bases de datos internacionales, es necesario encontrar aquellos modelos que muestren una mayor precisión de forma conjunta. El empleo de multiclasificadores (clasificadores que se fusionan para obtener un resultado mejor que los clasificadores individuales) ha sido una estrategia muy efectiva en la identificación o clasificación de medicamentos, que puede mejorar el poder predictivo de los modelos individuales. Para dar solución a este problema en el año 2013 fue desarrollado el software DASDE® 1.0 (Diversity Analysis, Selection and Discovery of best Ensemble from bases models) (Meneses Gómez & Cedré Gutiérrez 2013) por investigadores del CBQ de la UCLV.

El software DASDE versión 1.0 tiene varias funcionalidades, dentro de ellas combinar las salidas de varios modelos individuales de clasificación para obtener mejores predicciones. En las aplicaciones bioquímicas no solo existen grandes volúmenes de datos sino que además las bases de datos requieren de la aplicación de un número considerable de modelos individuales de clasificación por lo que se hace difícil la elección de cuáles modelos combinar de forma tal que el resultado que se obtenga supere el mejor obtenido por los clasificadores individuales, pues con tan solo un pequeño número de clasificadores se pueden generar grandes cantidades de combinaciones por lo que estamos frente a un problema combinatorio.

Otra de las funcionalidades del software DASDE versión 1.0 es la selección de las combinaciones de los modelos más diversos con el objetivo de combinar aquellos que no sean idénticos entre sí, pues entonces la combinación no arrojaría resultados superiores. La selección anterior se realiza mediante el uso de varias medidas estadísticas llamadas en la literatura “medidas de diversidad” (Kuncheva & Whitaker 2003), pero nuevamente estamos en presencia de un problema combinatorio por la misma razón, pues la existencia de un gran número de clasificadores individuales genera grandes cantidades de posibles combinaciones.

Ante la presencia de los dos problemas combinatorios descritos anteriormente en el software DASDE versión 1.0, se hace necesario aplicar una de las meta-heurísticas existentes pues la búsqueda exhaustiva de las soluciones provoca grandes demoras en las ejecuciones y también dependiendo de las capacidades de memoria de la máquina en varios casos no es posible encontrar soluciones.

Precisamente las meta heurísticas ofrecen la posibilidad de no tener que explorar todo el espacio de búsqueda de esas "posible soluciones". En trabajos recientes como (Morales 2014) se ha modelado este problema para otra herramienta computacional utilizando la meta heurística Algoritmos Genéticos (AG), ya que se adapta con facilidad a la problemática y se obtienen buenos resultados con la misma, por lo que se pretende en este trabajo aplicar dicha meta heurística y además ver sus resultados específicamente en el campo de la quimio-bioinformática, por todo lo anteriormente expuesto se enuncia el siguiente objetivo general:

Objetivo General

Incorporar a la herramienta computacional DASDE V1.0 la implementación de la meta heurística Algoritmo Genético para obtener soluciones eficientes y eficaces en aplicaciones quimio-bioinformática.

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

1. Analizar los conceptos básicos de las medidas de diversidad y de la meta heurística Algoritmos Genéticos.
2. Incorporar a la herramienta DASDE la implementación de las medidas de diversidad no pareadas: desacuerdo entre expertos y generalizada.
3. Incorporar a la herramienta computacional DASDE la implementación de la modelación de la meta heurística Algoritmos Genéticos para obtener combinaciones de los modelos más diversos.
4. Incorporar a la herramienta computacional DASDE la implementación de la modelación de la meta heurística Algoritmos Genéticos para obtener combinaciones de modelos que superen la clasificación individual.
5. Validar el desempeño de la meta heurística mediante la solución a dos problemas reales de quimio-bioinformática.

Se formularon las siguientes preguntas de investigación:

Preguntas de investigación:

1. ¿Cómo incorporar al software DASDE la implementación de la meta heurística AG para obtener combinaciones de clasificadores diversos?

2. ¿Cómo incorporar al software DASDE la implementación de la meta heurística AG para obtener combinaciones de modelos que superen la clasificación individual?
3. ¿Resolverá la incorporación de la meta heurística AG, de manera eficiente y eficaz problemas reales de quimio-bioinformática?

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

La incorporación de la meta heurística Algoritmos Genéticos al software DASDE permite obtener soluciones eficientes y eficaces en aplicaciones reales de la quimio-bioinformática.

La tesis está estructurada en tres capítulos. El primero contiene una descripción general de los sistemas de multiclasificadores. Además, se describen las medidas de diversidad existentes en la literatura y la meta heurística Algoritmo Genético donde se especifican sus características generales.

En el capítulo dos se describe las funcionalidades principales de la primera versión del software, luego se exponen las modificaciones realizadas a la herramienta, así como la modelación e implementación del Algoritmo Genético, finalmente se muestran algunos diagramas creados para comprender mejor el diseño del nuevo módulo incorporado a la herramienta.

En el capítulo tres se realiza un estudio con dos aplicaciones reales de la quimio-bioinformática. En estas, se usaron las funcionalidades incorporadas al software con la meta heurística AG, donde se obtienen buenos resultados, optimizando el tiempo de ejecución. Luego se realiza una comparación entre los resultados obtenidos y resultados previos obtenidos en (Meneses Gómez & Cedré Gutiérrez 2013). Finalmente se muestra el manual de usuario de las modificaciones incorporadas a la interfaz visual del software. Por último, se enuncian las conclusiones y recomendaciones del presente trabajo.

MULTICLASIFICACIÓN, MEDIDAS DE DIVERSIDAD Y ALGORITMOS GENÉTICOS

CAPÍTULO 1. MULTICLASIFICACIÓN, MEDIDAS DE DIVERSIDAD Y ALGORITMOS GENÉTICOS

En este capítulo se presenta la teoría relativa a los clasificadores y la multiclasificación. Además, se explican detalladamente las medidas de diversidad reportadas en la literatura para determinar cuán diverso es un grupo de clasificadores y por último se muestran los elementos principales relacionados con la meta heurística Algoritmos Genéticos.

1.1 Clasificadores. Modelos de construcción de multiclasificadores.

Los métodos de clasificación están caracterizados fundamentalmente porque se conoce la información de la clase a la que pertenece cada uno de los objetos. Cuando la variable de decisión, función o hipótesis a predecir es continua, a los algoritmos relacionados con los problemas supervisados se les conoce como métodos de regresión; y si es discreta, ellos se conocen como métodos de clasificación o simplemente clasificadores.(Bonet Cruz 2008).
De manera general, se puede decir que los métodos de clasificación son un mecanismo de aprendizaje, donde el objetivo es tomar cada instancia y asignarla a una clase en particular. La clasificación puede dividirse en tres procesos fundamentales: pre-procesamiento de los datos, selección del modelo de clasificación y, entrenamiento y prueba del clasificador (Bonet Cruz 2008).

De manera general, se puede decir que los métodos de clasificación son un mecanismo de aprendizaje y por lo tanto un proceso de inducción del conocimiento donde la tarea es tomar cada instancia y asignarla a una clase en particular.(Bello et al. 2001)

Entre los métodos de clasificación más usados están los algoritmos basados en casos, los árboles de decisión, las redes bayesianas, las redes neuronales artificiales, el análisis discriminante y la regresión logística.
Durante varios años han surgido un número considerable de métodos de clasificación, y aunque en algunos casos podamos reconocer uno de ellos como el mejor dotado para la tarea que pretendemos afrontar, no es lo habitual disponer de un clasificador perfecto que alcance una buena precisión. En otros muchos casos no podemos ni tan siquiera disponer de una lista ordenada que nos facilite el nombre del mejor sistema de clasificación. Por lo que ante la gran diversidad de diseños de clasificadores; si 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.

1.1.1 Modelos de construcción de multiclasificadores.

Muchos autores 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.

La combinación de clasificadores es un área activa de investigación en el aprendizaje automatizado y el reconocimiento de patrones. Se han realizado y publicado numerosos estudios teórico y empíricos que demuestran las ventajas de la combinación de clasificadores por encima de los modelos individuales (Kuncheva 2004).

Algunos aspectos que hacen que estos sistemas con varios clasificadores puedan ser mejores que un único clasificador, son:

- ✓ La decisión combinada mejora a las decisiones individuales.
- ✓ Los errores correlacionados de los clasificadores individuales pueden ser eliminados por medio de la combinación, considerándose el total de las decisiones.
- ✓ Los patrones de entrenamiento pueden no proporcionar la información suficiente para seleccionar el mejor clasificador.
- ✓ El algoritmo de aprendizaje puede no ser adaptado para solucionar el problema.
- ✓ El espacio individual de búsqueda puede no contener la función que se propone.

Además, Dietterich plantea tres razones para justificar la superioridad de la combinación de clasificadores sobre los individuales: razón estadística, razón computacional y razón representacional, las cuales se describen a continuación:

- ✓ **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 (Dietterich 2000)

- ✓ **Razón Computacional**

Aun 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 (Dietterich 2000) .

- ✓ **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 (Dietterich 2000).

Existen varias formas de construir multiclasificadores, hay una serie de algoritmos desarrollados, algunos diseñados para problemas generales y otros para problemas específicos, pero todos los algoritmos tienen como partes esenciales y fundamentales: la selección de los clasificadores de base y la elección de la forma de combinar las salidas (Bonet Cruz 2008).

Entre los modelos más populares que combinan clasificadores se encuentran: Bagging, Boosting, Stacking, métodos basados en rasgos y Vote, a continuación, se describen brevemente:

- ✓ Bagging: es uno de los primeros algoritmos de multiclasificación. Se basa en crear diferentes conjuntos de entrenamiento, extraídos del conjunto inicial de manera aleatoria y con remplazo, con lo cual asegura la diversidad. Este modelo necesita la selección de un modelo de clasificador inestable, o sea, un modelo que con pequeños cambios obtenga valores diferentes. Además usa un único modelo de clasificador y la combinación de los clasificadores resultantes se realiza con la técnica de voto mayoritario (Breiman 1996; Witten & Frank 2005).
- ✓ Boosting: Es parecido a Bagging porque usa el método de crear bases de entrenamiento aleatorias con reemplazo, a partir de la base original y un único modelo de clasificación para los clasificadores de base, de ahí que la diversidad la garantice de la misma forma. Sin embargo, este algoritmo se realiza de manera secuencial, donde los clasificadores se van entrenando uno detrás del otro porque usan información del anterior. Otra diferencia es que Boosting le da un peso al modelo por su rendimiento, en lugar de dar peso igual a todos los modelos. El

reemplazo se realiza estratégicamente de forma que los casos mal clasificados tienen mayor probabilidad, que los bien clasificados, de pertenecer al conjunto de entrenamiento del siguiente clasificador del sistema (Schapire 1990).

- ✓ Stacking: Es un método diferente a los anteriores pues la diversidad se determina con el empleo de diversos modelos de clasificación. Es menos utilizado que Bagging y Boosting, ya que es difícil de analizar teóricamente. Stacking combina múltiples clasificadores generados por diferentes algoritmos para un mismo conjunto de datos en una primera fase. Para combinar las salidas no utiliza voto mayoritario, sino que introduce un meta clasificador que aprende la relación entre las salidas de los clasificadores de base y la clase original (Wolpert 1992).
- ✓ Métodos basados en rasgos: En la construcción de un multiclasificador, los clasificadores de base pueden ser obtenidos a partir de subconjuntos de rasgos diferentes, lo cual es otra forma de buscar diversidad. La selección de rasgos tiene como objetivo lograr una mayor eficiencia en los cálculos, así como una mayor exactitud del multiclasificador. De esa manera puede que los clasificadores individuales no sean tan precisos o exactos, pero sí sean más diversos. Existen muchos modelos de multiclasificadores que utilizan subconjuntos de rasgos diferentes como los descritos por Kuncheva en (Kuncheva 2004).
- ✓ Vote: Al igual que Stacking, establece la diversidad con la utilización de diferentes modelos de clasificación como clasificadores base. Las salidas de estos clasificadores están dadas por vectores con una distribución de probabilidad para cada una de las clases. Vote combina estas probabilidades utilizando diferentes criterios como voto mayoritario, promedio, mínimo, máximo o mediana de las probabilidades.

1.1.2 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 & Jain 2000; Bulacio 2006). La estrategia a utilizar en este trabajo está basada en la estrategia de fusión.

- ✓ Fusión de clasificadores

La fusión de clasificadores asume que todos los clasificadores son competitivos y complementarios (igualmente *expertos*). Por eso, 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).

En la literatura se refieren a otras estrategias de fusión, por ejemplo, en (Segrera & Moreno 2006) se proporcionan tres estrategias de fusión, 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

1.1.3 Algunas investigaciones en la combinación de clasificadores.

La combinación de clasificadores ha cobrado gran auge en la actualidad como se mencionaba anteriormente en otros epígrafes, en algunas investigaciones como en (Rahman & Tasnim 2014) se plantean el uso de diversos métodos de generación de combinaciones de clasificadores, los cuales se pueden clasificar en seis grupos que se basan en la manipulación de la formación de parámetros, los grupos son los siguientes: la manipulación de la función de error, la manipulación de la función de espacio, la manipulación de las etiquetas de salida, la agrupación, y la manipulación de los patrones de entrenamiento.

A continuación, se muestran ejemplos de algunas investigaciones recientes alrededor de este tema:

- ✓ En el artículo *A Bound on Kappa-Error Diagrams for Analysis of Classifier Ensembles* se plantea el uso de los diagramas kappa-error para obtener información acerca de por qué un método de ensamblado de clasificadores es mejor que otro en un determinado conjunto de datos. Un punto en el diagrama corresponde a un par de clasificadores. El eje *x* es la diversidad de pares (kappa), y el eje *y* es el error individual promediado. En este estudio dicho diagrama se utilizó para conseguir puntos de vista en la comparación de conjuntos de clasificadores, kappa se calcula a partir de la matriz de contingencia 2x2 correcta / incorrecta. Se deriva una cota inferior de kappa que determina la parte viable del diagrama kappa-error. Las simulaciones y experimentos con datos reales demostraron que hay espacio factible en el diagrama correspondiente a los mejores conjuntos, y que la precisión del individuo es el factor principal en la mejora de la exactitud. (Kuncheva 2013)
- ✓ En el artículo *A weighted voting framework for classifiers ensembles* proponen un marco probabilístico para la combinación de clasificadores, lo que da rigurosas condiciones de optimización (error de clasificación mínima) para cuatro métodos de combinación: voto mayoritario (MV), voto mayoritario ponderado (WMV),

r*ecall* (REC) y *Naive Bayes* (NB). Este marco se basa en dos partes: la independencia de clase condicional de las salidas de los clasificadores y una suposición acerca de las precisiones individuales. Los resultados experimentales revelan que no hay un mejor combinador definitivo entre los cuatro métodos candidatos, dando una ligera preferencia a NB, este combinador resultó mejor para los problemas con un gran número de clases debido balance, mientras que WMV resulta mejor para los problemas con un pequeño número de clases no balanceadas. Otra dirección para futuras investigaciones planteada en este artículo es: Cómo los combinadores están influenciados por: la elección de la base de clasificador, los tamaños de conjunto, el método de producción de las amplificaciones de base y la diversidad del conjunto. (Kuncheva & Rodríguez 2014)

- ✓ En el artículo *Random Balance: Ensembles of variable priors classifiers for imbalanced data* se propone un nuevo enfoque para la construcción de conjuntos de clasificadores para dos clases de datos desequilibradas, llamadas *Random Balance* Donde cada miembro del conjunto es entrenado con datos incluidos en la muestra del conjunto de entrenamiento y aumentados por las instancias artificiales obtenidas una vez aplicado *SMOTE*. Lo novedoso en este enfoque es que las proporciones de las clases para cada miembro del conjunto se eligen al azar. Los experimentos realizados en este trabajo conllevan a la proposición de un nuevo método de ensamble de clasificadores llamado *RB-Boost* que combina *Random Balance* con *AdaBoost.M2*. Esta combinación involucra a implementar al azar proporciones de clase, además de instancia en una nueva ponderación. Además de impulsar las AUC, esta heurística intuitiva evita la necesidad de ajustar el parámetro proporción clase sensible, común a la mayoría de los métodos de clasificación desequilibrado. A pesar de su simplicidad, los métodos propuestos han demostrado ser competitivo en comparación con otros conjuntos del estado de la técnica, incluyendo los creados específicamente para la clasificación de datos desequilibrado.(Díez-pastor et al. 2015).
- ✓ En el artículo *Random Forests* se plantea que dicho algoritmo es una combinación de árboles predictivos donde cada árbol depende del valor aleatorio de un vector independientemente y tiene la misma distribución que todos los arboles del *forests*. El error de generalización del *forests*, así como el límite de la cantidad de árboles se hace grande. El error de generalización del *forests* depende de la fortaleza del árbol

de clasificación individuales, así como la correlación entre ellos. El uso de una selección aleatoria de la razón del error de cada campo en los nodos de la división es comparado favorablemente por *Adaboost*, pero son más robusto con respecto al ruido. estimaciones internas del error, la fortaleza, y la correlación son usadas para mostrar el incremento del número de características en la división. Los cálculos internos se utilizan para medir la importancia de la variable. Estas ideas son también aplicadas en la regresión.(Breiman 2001)

1.2 Medidas de diversidad como criterio para seleccionar los modelos.

La diversidad entre los clasificadores de base es muy importante, ya que de esto dependerá en gran medida el resultado final del multiclasificador. La diversidad de los errores de los clasificadores puede dar una medida del mayor valor posible que se puede aspirar con la combinación de esos modelos. Como ya se discutió anteriormente, algunos multiclasificadores, como *Bagging* y *Boosting*, garantizan la diversidad utilizando diferentes conjuntos de bases de entrenamiento. Otros utilizan diferentes conjuntos de rasgos y otros diferentes clasificadores de base. En estos dos últimos casos no se garantiza una gran diversidad, por lo que es preciso el uso de algunas medidas estadísticas que permitan hacer estimación de cuán diversos son los clasificadores.(Kuncheva 2004)

En (Kuncheva & Whitaker 2003) 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. La elección de la medida a utilizar va a depender directamente de la cantidad de clasificadores a utilizar y de muchos otros aspectos que deben ser estudiados con detalle en investigaciones futuras. La mayoría de estas medidas son estadísticas.

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

Las medidas de diversidad que se basan en todo el conjunto consideran la salida de todos los clasificadores a la vez y calculan un único valor de diversidad para todo el conjunto. A continuación, se muestran en la Figura 1.2 dichas medidas.

1.2.1 Medidas de diversidad en forma de pares (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. Ellas son: desacuerdo, doble falta, Q estadístico, y Coeficiente de Correlación.

A continuación, se indica el resultado de dos clasificadores (C_i, C_j) para una instancia en cuanto a la clasificación correcta o no.

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

Tabla 1.1 Clasificación entre los resultados de los clasificadores Ci y Cj para una instancia

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:

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

Tabla 1.2 Clasificación entre los resultados de los clasificadores Ci y Cj para todas las instancias

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.

En la Figura 1.1 Medidas de diversidad en forma de pares.

A continuación, se muestra una breve explicación de cada una de ellas:

se muestran las medidas de diversidad en forma de pares.

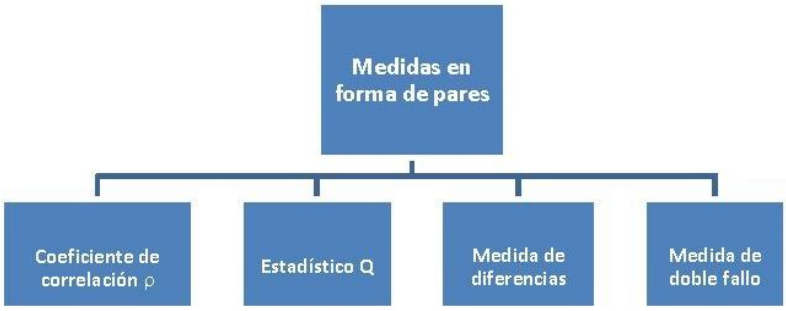

Figura 1.1 Medidas de diversidad en forma de pares.

A continuación, se muestra una breve explicación de cada una de ellas:

1.2.1.1 Medida de desacuerdo

Enunciada por Skalak en (Skalak 1996) 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.

$$D_{ij} = \frac{B+C}{N} \quad Donde\ D_{ij} \in [0,1] \qquad \text{Eq. 1.1}$$

1.2.1.2 Medida de doble falta

Otra de las medidas se conoce como medida de doble falta introducida por Giacinto y Roli (Giacinto & Roli 2001), considera el fallo de los dos clasificadores al mismo tiempo. Ruta y Gabrys (Ruta & 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.

$$DF_{ij} = \frac{D}{N} \qquad DF_{ij} \in [0,1] \qquad \text{Eq. 1.2}$$

1.2.1.3 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 & Whitaker 2003):

$$\rho_{ij} = \frac{A * D - B * C}{\sqrt{(A+B) * (C+D) * (A+C) * (B+D)}} \qquad \text{Eq. 1.3}$$

El coeficiente de correlación cuenta con las siguientes características:

a) El valor del resultado 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.

1.2.1.4 Q-estadístico

Estadístico usado para evaluar similitud entre clasificadores (Yule 1990), se calcula por la siguiente formula:

$$Q_{ij} = \frac{A * D - B * C}{A * D + B * C} \quad Q \in [-1; 1] \qquad \text{Eq. 1.4}$$

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

1.2.2 Medidas de diversidad para todo el conjunto (non-pairwise)

En la Figura 1.2 Medidas de diversidad grupales se muestran varias de las medidas de diversidad grupales.

Figura 1.2 Medidas de diversidad grupales

1.2.2.1 Entropía

La medida de Entropía (TheEntropyMeasure) (Kuncheva & Whitaker 2003) se basa en la idea intuitiva de que en un conjunto de N casos y L clasificadores la mayor diversidad se obtendrá si L/2 de los clasificadores clasifican una instancia correctamente y los otros L-L/2 la clasifican incorrectamente. Fue introducida por (Cunningham & Carney 2000)

$$E = \frac{1}{N} \cdot \frac{2}{L-1} \sum_{j=1}^{N} min\left\{\left(\sum_{i=1}^{L} y_{j,i}\right), \left(L - \sum_{i=1}^{L} y_{j,i}\right)\right\} \quad , y_{j,i} \in \{0,1\}, \;\; 0 \leq E \leq 1 \qquad \text{Eq 1.5}$$

Donde $y_{j,i}$ tendrá valor 1 si el clasificador i clasificó correctamente el caso j y 0 en caso contrario. Si E tiene valor 0 esto indica que no hay diferencia entre los clasificadores y un valor 1 indica la mayor diversidad posible.

1.2.2.2 Varianza de Kohavi-Wolpert

La varianza de Kohavi-Wolpert (Kohavi-WolpertVariance), fue inicialmente propuesta por (Kohavi & Wolpert 1996) Esta medida es originada de la descomposición de la varianza del sesgo del error de un clasificador.

Kuncheva y Whitaker presentaron en (Kuncheva & Whitaker 2003) una modificación para medir la diversidad de un ensamblado compuesto por clasificadores binarios, quedando la medida de diversidad como:

$$KW = \frac{1}{NL^2} \sum_{j=1}^{N} Y(z_j) \left(L - Y(z_j)\right) \text{ , } 0 \leq \text{KW} \leq 1 \;\; donde Y(z_j) = \sum_{i=1}^{L} y_{i,j} \qquad \text{Eq. 1.6}$$

Con esta medida, la diversidad disminuye a medida que el valor de KW aumenta.

1.2.2.3 Medida de desacuerdo entre expertos

La medida de desacuerdo entre expertos (Measurement Interrater Agreement) (Fleiss 1981) se desarrolla como una medida de fiabilidad entre clasificadores. Puede usarse para medir el nivel de acuerdo dentro de un conjunto de clasificadores, por consiguiente, está también basada en el supuesto que un conjunto de clasificadores debe discrepar entre sí para ser diverso. La diversidad disminuye cuando el valor de k aumenta. El k se calcula por:

$$k = 1 - \frac{\frac{1}{L}\sum_{j=1}^{N} Y(Z_j)\left(L - Y(Z_j)\right)}{N(L-1)p(1-p)}, -1 \leq k \leq 1 \qquad \text{Eq. 1.7}$$

Donde el término de la derecha es la medida de concordancia de Kendall y p es la media de la exactitud de la clasificación individual, y se calcula como:

$$p = \frac{1}{N \cdot L} \cdot \sum_{j=1}^{N} \sum_{i=1}^{L} y_{j,i} \qquad \text{Eq. 1.8}$$

1.2.2.4 Medida de dificultad

La medida de dificultad (The Measure of "difficulty") viene del estudio realizado por Hansen y Salamon (Hansen & Salomon 1990). Se calcula a través de la varianza de una variable aleatoria discreta que toma valores en el conjunto $\{0/L, 1/L, 2/L, \dots, 1\}$ y denota la probabilidad de que exactamente *i* clasificadores hayan clasificado bien todas las instancias.

Para conveniencia, θ suele ser escalada linealmente en el intervalo [0,1] tomando $p(1-p)$ como el mayor valor posible, donde p es la precisión individual de cada clasificador. La diversidad del ensamblado aumenta con el decremento del valor de la medida de dificultad.

La intuición de esta medida puede ser explicada de la siguiente manera: un ensamblado de clasificadores diverso tiene un valor pequeño de medida de dificultad, dado que cada muestra de entrenamiento puede al menos ser clasificada correctamente por una proporción de todos los clasificadores base, lo cual es más probable con una baja varianza de *X*.

$$\theta = Var(x) \qquad \text{Eq. 1.9}$$

1.2.2.5 Medida de diversidad generalizada

La medida de diversidad generalizada (Generalized Diversity) se enunció por Partridge y Krzanowski (Partridge & Krzanowski 1997).

Sea *Y* una variable aleatoria que representa la proporción de clasificadores que clasificaron incorrectamente una muestra $x \in R^n$ extraída aleatoriamente del conjunto de datos. Denotemos por pi la probabilidad de que $Y = i/L$ y $p(i)$ la probabilidad de que *i* clasificadores extraídos de manera aleatoria fallen en clasificar correctamente un objeto *X* extraído aleatoriamente. Supongamos que dos clasificadores son tomados de forma aleatoria del ensamblado *D*, Partridge y Krzanowski exponen en su trabajo que la máxima diversidad es lograda cuando uno de los dos clasificadores se equivoca en clasificar un objeto y el otro lo clasifica correctamente. En este caso la probabilidad de equivocarse los dos clasificadores es $p(2) = 0$. Por otra parte argumentan que la mínima diversidad se lograría cuando el fallo de un clasificador es siempre acompañado con el fallo del otro, entonces la probabilidad de que los dos clasificadores fallen es la misma que la probabilidad de que un clasificador escogido de forma aleatoria falle, esto es $p(1)$.

$$GD = 1 - \frac{p(2)}{p(1)}, 0 \le GD \le 1, \text{ donde} \qquad \text{Eq. 1.10}$$

$$p(1) = \sum_{i=1}^{L} \frac{i}{L} * p[i], \quad p(2) = \sum_{i=1}^{L} \frac{i*(i-1)}{L*(L-1)} * p[i] \qquad \text{Eq. 1.11}$$

El valor de *GD* varía entre 0 y 1, siendo 0 la menor diversidad cuando $p(2) = p(1)$ y 1 la mayor diversidad cuando $p(2) = 0$ y L la cantidad de clasificadores.

1.2.2.6 Medida de diversidad de coincidencia de fallos

Esta medida (CoincidentFailureDiversity) se enuncia por Partridge y Krzanowski (Partridge & Krzanowski 1997), como una mejora a la medida anterior.

Esta medida está diseñada de tal forma que tenga un valor mínimo 0 cuando todos los clasificadores siempre clasifiquen correctamente o cuando todos los clasificadores lo mismo clasifiquen correcta o incorrectamente al mismo tiempo. Su máximo valor 1 es alcanzado cuando todos los errores de clasificación son únicos, es decir cuando al menos un clasificador va a clasificar incorrectamente cualquier objeto aleatorio.

$$\boldsymbol{CFD} = \begin{cases} \mathbf{0}, & \boldsymbol{p}[\mathbf{0}] = \mathbf{1} \\ \frac{\mathbf{1}}{\mathbf{1}-\boldsymbol{p}[\mathbf{0}]} * \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{L}} \frac{\boldsymbol{L}-\boldsymbol{i}}{\boldsymbol{L}-\mathbf{1}} \times \boldsymbol{p}[\boldsymbol{i}], & \boldsymbol{p}[\mathbf{0}] < \mathbf{1} \end{cases}$$

Eq. 1.12

p[0]=1 cuando todos los clasificadores siempre son simultáneamente correctos o incorrectos, p[i] es el mismo término de la medida anterior y L es la cantidad de clasificadores. El valor de CFD está entre 0 y 1 y mientras mayor sea, mayor será la diversidad.

1.2.2.7 Medida de diversidad de distintos fallos.

Esta medida (DistinticFailureDiversity) fue igualmente enunciada por Partridge y Krzanowski (Partridge & Krzanowski 1997), como una mejora a la medida anterior, pues ahora se va a tener en cuenta todas las instancias donde los clasificadores no coinciden en las clases asignadas, es decir, se consideran las distintas posibilidades de fallo teniendo en cuenta las clases.

$$DFD = \begin{cases} 0, & t[i] = 0 \\ \sum_{i=1}^{L} \frac{L-i}{L-1} \times t[i], & t[i] > 0 \end{cases} \qquad \text{Eq. 1.13}$$

Aquí *t* es un vector de probabilidades en el que cada posición se calcula determinando la cantidad de *i* clasificadores que hayan fallado en asignar la clase correcta dividido por el total de fallos ocurridos y L es la cantidad de clasificadores. El valor de DFD está entre 0 y 1 y mientras mayor sea, mayor será la diversidad.

1.2.2.8 Medida de variabilidad

Esta medida (TheMeasureofVariability*)* tiene en cuenta si las clases asignadas por los clasificadores en cada instancia son distintas o no. Mientras mayor sea su valor, mayor será la diversidad.

$$Var = \frac{\sum_{i=1}^{N} a}{N} \; dondea = \begin{cases} 0 & si E_1(i) = E_2(i) = \cdots = E_L(i) \\ 1 & e.o.c \end{cases} \qquad \text{Eq. 1.14}$$

Donde *N* es el total de instancias y$E_L(i)$ es la etiqueta (clase) asignada a la instancia *i*, por el clasificador *i*-ésimo.

1.2.3 Algunas investigaciones sobre las medidas de diversidad.

A continuación, se muestran algunas de las investigaciones recientes donde se usan las medidas de diversidad:

- ✓ En el artículo *"Good" and "Bad" Diversity in Majority Vote Ensembles* se plantea que a pesar de que la diversidad en la combinación de clasificadores es deseable, su relación con la precisión del conjunto obtenido no es sencillo. Los autores derivan una descomposición del error del voto mayoritario en tres términos: la media de precisión individual, la "buena" diversidad y la "mala" diversidad. El término de "buena" diversidad se extrae del error individual del voto mientras que el término de "mala" diversidad se añade al error individual. Un valor mayor del primero reduce el error de la mayoría de votos, mientras que un valor mayor del segundo aumenta el error. Se relacionan los dos términos de diversidad a los límites de voto mayoritario definidos (los patrones de éxito y fracaso). En el trabajo se adopta la perspectiva de que una medida de la diversidad debe ser de origen natural, como consecuencia directa de dos factores: la función de pérdida de interés, y la función de combinación. Han demostrado que existe una relación directa de estos conceptos a los límites superiores / inferiores definidos por el error de la votación de la mayoría. El estudio de simulación ilustró que los términos de diversidad tienden a mostrar una gran variación en conjuntos más pequeños, y estabilizarse con grandes conjuntos. Los resultados de este trabajo sugieren que puede ser posible construir algoritmos más específicos, que magnifican directamente a la "buena" la diversidad, mientras que supriman el término de "mala" diversidad.(Brown & Kuncheva 2010)
- ✓ En el artículo *Diversity measures for one-class classifier ensembles* plantea que la clasificación de una clase, es uno de los temas más desafiantes en *machine learning*

y poca la atención para la tarea de crear eficientes conjuntos de una clase. En este trabajo se propone un nuevo modelo dedicado para la clasificación de una clase e introducen nueva medida de diversidad dedicada a este. Los modelos de medidas pensadas y de clasificación fueron evaluados con base en experimentos de la computadora que fueron efectuados en un conjunto diverso, de estándar de comparación. Los resultados confirmaron que la introducción de las medidas de diversidad dedicadas a conjuntos de una clase es una dirección de investigación importante y prueban que los modelos pensados son proposiciones valiosas que pueden funcionar mejor que los métodos tradicionales (Krawczyk & Woźniak 2014)

- ✓ En (Santana et al. 2006) se afirma que el ensamble de clasificadores es una forma efectiva de perfeccionar el desempeño de los clasificadores individuales. En este escrito se presenta un método basado en DCS (*Dynamic Classifier Selection*), teniéndose en cuenta desempeño y diversidad de los clasificadores para escoger los miembros del ensamble. Se afirma que la situación ideal es cuando se presentan en el multiclasificador clasificadores que presentan errores no correlacionados, donde el conjunto debe mostrar diversidad entre los miembros para mejorar el desempeño de los clasificadores individuales. Se describe que la diversidad puede ser cumplida en tres formas diferentes: las variaciones de los parámetros del clasificador, variaciones de la data de entrenamiento del clasificador, variación del tipo de medidas de diversidad del clasificador.

1.3 Procedimientos meta heurísticos

Los algoritmos heurísticos intentan encontrar la solución de algún problema sin necesidad de tener que explorar todo el espacio de soluciones, son muy usados en el ámbito de la optimización.

De manera práctica, optimizar significa poco más que mejorar. Sin embargo, científicamente se traduce como el proceso de encontrar la mejor solución posible para una determinada problemática. En un problema de optimización existen distintas soluciones y un criterio para discriminar entre ellas. De forma más precisa, este problema consiste en encontrar el valor de una variable de decisión para las cuales una determinada función objetivo alcanza su valor máximo o mínimo.

En (Fernández et al. 1996) se plantea que un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva,

en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

En contraposición a los métodos de búsqueda exhaustiva que proporcionan una solución óptima del problema, los métodos heurísticos se limitan a obtener una buena solución, no necesariamente esta solución es la óptima. Por lo que, el tiempo invertido por un método exacto para encontrar el óptimo de un problema complejo es de un orden de magnitud muy superior al del heurístico (pudiendo llegar a ser tan grande que en muchos casos resulta inaplicable).

Existen varias razones para utilizar métodos heurísticos, entre las que podemos mencionar:

- ✓ El problema es de una naturaleza tal que no se conoce ningún método exacto para su resolución.
- ✓ Aunque existe un método exacto para resolver el problema, su uso es computacionalmente muy costoso.
- ✓ El método heurístico es más flexible que un método exacto, permitiendo, por ejemplo, la incorporación de condiciones de difícil modelación.
- ✓ El método heurístico se utiliza como parte de un procedimiento global que garantiza el óptimo de un problema. Existen dos posibilidades:

- El método heurístico proporciona una buena solución inicial de partida.
- El método heurístico participa en un paso intermedio del procedimiento.

Existen muchos métodos heurísticos entre los más relevantes se destacan métodos de descomposición, inductivos, de reducción, constructivos y de búsqueda local. Si bien todos estos métodos han contribuido a ampliar el conocimiento para la resolución de problemas reales, los métodos constructivos y los de búsqueda local constituyen la base de los procedimientos metaheurísticos.

Según (Osman & Kelly 1996), los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria. Estos procedimientos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la Inteligencia Artificial, la evolución biológica y los mecanismos estadísticos.

Entre este tipo de algoritmos, por su eficacia sobre una colección significativa de problemas, sobresalen la búsqueda Tabú (Glover & Melián-Batista 2003), el recocido

simulado (Dowsland & Adenso-Díaz 2003) y los métodos inspirados en procesos naturales que incluyen la optimización basada en colonia de hormigas (ACO) (Dorigo & Birattari 2010), optimización por enjambre de partículas (Kennedy 2010) y los Algoritmos Genéticos (AG) (Goldberg 1989). Este último ha probado su eficacia sobre un gran número de problemas y provee una fácil modelación para nuestra problemática explicada en la introducción, por lo que, posteriormente es profundizado con más detalle en este trabajo.

1.3.1 Los Algoritmos Genéticos.

Los métodos evolutivos están basados en poblaciones de soluciones. A diferencia de otros métodos, en cada una de las iteraciones del algoritmo no se obtiene una única solución sino un conjunto de éstas. Estos métodos se basan en generar, seleccionar, combinar y reemplazar un conjunto de soluciones. Dado que mantienen y trabajan un conjunto en lugar de una única solución en todo el proceso de búsqueda, suelen presentar tiempos de computación un poco más altos que los de otras meta heurísticas. Esto se ve agravado por la convergencia de la población que requiere de un gran número de iteraciones como es el caso de los Algoritmos Genéticos.

Los AG fueron introducidos por John Holland en 1970 inspirándose en el proceso evolutivo natural de los seres vivos.

Aunque muchos aspectos están todavía por discernir, existen principios generales de la evolución biológica aceptados por la comunidad científica (Martí 2002). Algunos de éstos son:

- ✓ La evolución opera en los cromosomas en lugar de en los individuos a los que representan.
- ✓ La selección natural es el proceso por el cual los cromosomas con “buenas estructuras” se reproducen más a menudo que los demás.
- ✓ En el proceso de reproducción tiene lugar la evolución mediante la combinación de los cromosomas de los progenitores. Entiéndase por recombinación al proceso en el que se forma el cromosoma del descendiente.
- ✓ Existen mutaciones que pueden alterar la información genética.
- ✓ La evolución biológica no tiene memoria pues en la formación de los cromosomas únicamente se considera la información del periodo anterior.

Los AG establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural. Cada uno de los cromosomas

(individuos) en la población tiene asociado un valor de una función, denominada calidad ("*fitness*") y esta se basada en la función objetivo del problema; la cual determina cuáles cromosomas son utilizados para formar nuevos cromosomas en el proceso evolutivo, conociéndose este proceso como *selección natural.* Ellos son creados usando operadores genéticos como el *cruzamiento* y la *mutación.*

La mayoría de los especialistas en esta rama coinciden en que los AG pueden resolver las dificultades representadas en los problemas de la vida real que a veces son imposibles de resolver por otros métodos. Según (Goldberg 1989), el tema fundamental y central de la investigación en AG consiste en la robustez: el balance entre la eficacia y la eficiencia necesaria para persistir en muchos ambientes diferentes.

Goldberg resalta las diversas formas en que se diferencian los AG de los sistemas tradicionales ejemplo de ello es que:

- ✓ Trabajan con una codificación del conjunto de parámetros, no con los parámetros en sí.
- ✓ Realizan la búsqueda a partir de una población de puntos, no de un punto simple.
- ✓ Sólo utilizan la información de la función objetivo, sin derivadas u otro conocimiento auxiliar.
- ✓ Utilizan reglas de transición probabilísticas, no determinísticas.

Este tipo de algoritmo en general tiene gran éxito en la búsqueda y optimización combinatoria. La razón de esto es, en gran parte, por sus posibilidades de explorar la información acumulada de un espacio de búsqueda desconocido con el objetivo de producir subespacios de búsqueda más útiles.

1.3.2 Visión general de los AG

Un AG comienza con una población de cromosomas generada aleatoriamente o establecida de antemano. Durante iteraciones sucesivas, denominadas generaciones, los cromosomas en la población son evaluados para su adaptación como solución, y sobre la base de esta evaluación, una nueva población de cromosomas es formada usando un mecanismo de selección y operadores genéticos específicos tales como el cruzamiento y la mutación. La calidad o función de evaluación de cada cromosoma, debe ser establecida para cada problema. Dado un cromosoma X, esta función de evaluación devuelve un valor numérico, el cual se supone que sea proporcional a la utilidad o adaptación de la solución que el cromosoma representa.

A continuación, se muestra el esquema tradicional de los AGs.

```
Procedure Algoritmo Genético
begin (1)
        t = 0;
        inicializar P(t);
        evaluar P(t);
        While (Not condición-acabado) do
        Begin (2)
                t = t + 1;
                seleccionar P(t) de P(t - 1);
                recombinar P(t);
                evaluar P(t);
        end (2)
end (1)
```

1.3.3 Técnicas de representación.

La representación del problema a resolver es el primer paso y el más importante en la implementación de un AG y de cualquier otro algoritmo de optimización según (Koza 1992). Cadenas de tamaño fijo y de codificación binaria para la representación de las posibles soluciones han sido usadas en los AGs desde los primeros resultados teóricos que muestran la efectividad de estos algoritmos (Goldberg 1990). Así, por ejemplo, para representar los elementos de un espacio de búsqueda $S = S_1 x... S_n$ usando el alfabeto binario, una función cod_i: $S_i \rightarrow \{0,1\}^{L_i}$, $L_i \in N$ es empleada para codificar cada elemento en S_i usando una cadena binaria de tamaño L_i.

No obstante, las potencialidades de los AG no radican solamente en estas representaciones (Antonisse 1989; Radcliffe 1992). Es por esto que en muchos casos se han propuesto representaciones no binarias, muchas de las cuales resultan más naturales para las aplicaciones de problemas específicos. Algunas de ellas son:

- ✓ Vector de números con punto flotante para problemas quimio métricos, optimización de funciones numéricas, diseño óptimo de filtros de múltiples capas, controladores de lógica difusa, etc.
- ✓ Vectores de números enteros para optimización de funciones, diseño paramétrico de aeronaves, aprendizaje no supervisado de redes neuronales, etc.
- ✓ Listas ordenadas para problemas de optimización de esquemas, problema del viajero vendedor, problemas de *job-shop scheduling*, etc.
- ✓ Expresiones *lisp* para el desarrollo de programas de computación para el control de tareas, planeación robótica y regresión simbólica.

1.3.4 Mecanismo de Selección.

Considérese una población *P* con cromosomas *C1*, *C2,…*, *CN*. El mecanismo de selección produce una población intermedia, *Pt* con copias de los cromosomas en *P*. El número de copias recibida por cada cromosoma depende de su calidad; cromosomas de mayor calidad tendrán más oportunidad de ser copiados en *Pt*. El mecanismo de selección consiste en dos pasos, el cálculo de la probabilidad de seleccionar un cromosoma y el algoritmo de selección.

i. Cálculo de la probabilidad de selección

Por cada cromosoma *Ci* en la población *P*, se calcula la probabilidad *ps(Ci)* de ser seleccionado. El mecanismo de selección más conocido usa la selección proporcional (Goldberg 1989; Holland 1975) donde *ps(Ci), i=1,…, N,* es calculada como:

$$p_s(C_i) = \frac{f(C_i)}{\sum_{i=1}^{N} f(C_i)} \qquad \text{Eq. 1.15}$$

Donde $f(C_i)$ es el valor de la función objetivo para el cromosoma i e $p_s(C_i)$ indica la probabilidad de que el *Ci* sea seleccionado para participar en el proceso de recombinación. De esta forma, los cromosomas con una calidad cercana al promedio anterior tienen más probabilidad de ser incluidos en la población intermedia.

Algoritmo de selección

El algoritmo de selección que se escoja debe seleccionar, apoyado en las probabilidades calculadas anteriormente, los cromosomas que formarán parte de la población intermedia y a los que se les aplicará la recombinación genética. El algoritmo clásico es el *muestreo probabilístico con reposición,* mejor conocido como el método de la ruleta (Goldberg 1989; Holland 1975). La población es mapeada en una ruleta, donde cada cromosoma *Ci* es representado por un espacio que es proporcional a *ps(Ci).* Mientras quede capacidad en P^t, un número aleatorio es generado y es seleccionado el cromosoma del espacio al que corresponde dicho número. La Figura 1.3 Aplicación de la selección usando el muestreo probabilístico con reposición muestra un ejemplo de la aplicación de este mecanismo de selección.

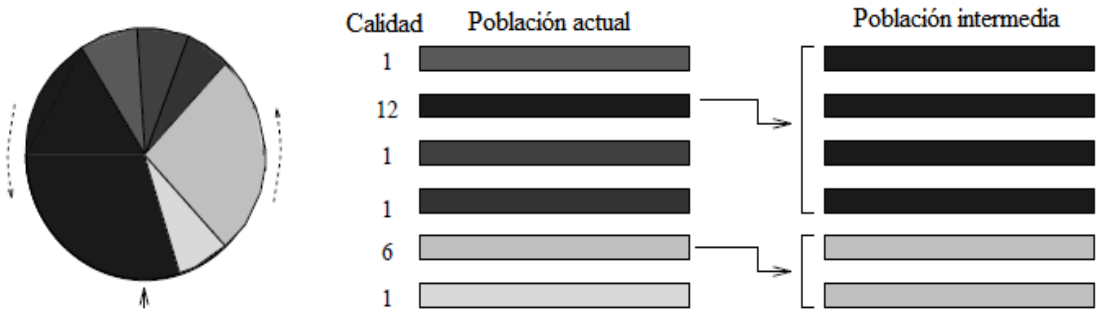

Figura 1.3 Aplicación de la selección usando el muestreo probabilístico con reposición

Pero también existen otros métodos de generar la población intermedia como:

Selección basada en el rango: Usa un procedimiento de ranqueo y los cromosomas son seleccionados usando la distribución de probabilidad

$$p([k]) = \frac{2k}{n(n+1)} \qquad \text{Eq. 1.16}$$

Donde [k] es el *k*-ésimo cromosoma ordenado en forma ascendente y n es el tamaño de la población.

Selección de torneo: Una alternativa que combina la idea anterior con el mecanismo de selección. Un cromosoma necesita ganar una competencia entre un conjunto de cromosomas seleccionados anteriormente.

Técnica del resto estocástico: Esta técnica establece que la cantidad de copias que debe tener una cadena específica debe ser calculada del siguiente modo; primero, de forma determinística se toma la parte entera del valor esperado de copias para esa cadena y segundo, la parte fraccionaria participa en una selección estocástica (la ruleta). De esta forma, la aleatoriedad se restringe sólo a la parte fraccionaria.

Elitismo: Establece que el mejor individuo o los mejores de la población sobrevivan de generación en generación. La estrategia de elitismo básica copia el mejor individuo de la población actual a la próxima si éste individuo no ha sido transferido a través del proceso normal de selección, cruzamiento y mutación.

1.3.5 Recombinación a través de los operadores de cruzamiento y mutación

Una vez realizada la selección de los cromosomas que van a ser recombinados, los operadores de cruzamiento y mutación son aplicados.

1.3.5.1 Operador de cruzamiento

El operador de cruzamiento es un método utilizado para compartir la información entre cromosomas. Combina los rasgos de dos cromosomas padres para formar dos hijos, con la posibilidad de que mejores cromosomas puedan ser generados. El operador de cruzamiento no es comúnmente aplicado a todos los pares de cromosomas en la población intermedia. Este operador es aplicado dependiendo de un número aleatorio generado que se compara con la probabilidad de que ocurra el cruzamiento. Las definiciones de este operador son muy dependientes de la representación del problema que se escoja. En el caso de la representación de cadena de bits, algunas definiciones pueden ser:

Operador clásico: Es el operador de cruzamiento clásico, presentado en (Holland 1975; Goldberg 1989). Dado dos cromosomas $C_1 = (C_1 \dots C_L^1)$ y $C_2 = (C_1^2 \dots C_L^2)$ son generados los hijos $H_1 = \{c_{1,\dots}^1, C_i^1, C_{i+1}^2, \dots, C_L^2$ y $H_2 = (C_1^2, \dots, C_i^2, C_{i+1}^1, \dots, C_L^1)$, donde i es un número aleatorio generado perteneciente a$\{1, \dots, L-1\}$. La Figura 1.4 Cruzamiento clásico en un punto muestra un ejemplo de la aplicación de este operador.

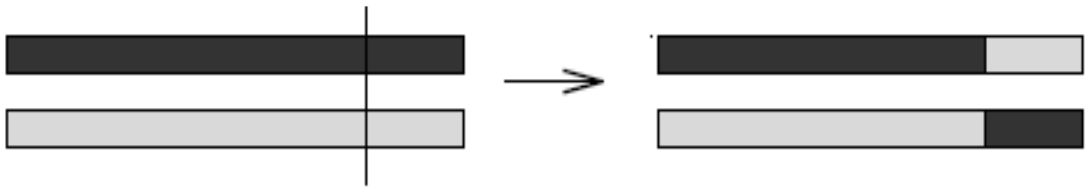

Figura 1.4 Cruzamiento clásico en un punto

Cruzamiento en *n* puntos: Es una generalización del cruzamiento anterior; *n* puntos de cruce son seleccionados aleatoriamente y los segmentos de los padres, definidos por dichos puntos, son intercambiados para generar los hijos (Eshelman et al. 1989).

Cruzamiento uniforme: Solo se obtiene un hijo, en vez de los dos que se obtenían en los cruzamientos anteriores. Los valores de cada gen en el hijo son determinados por una elección aleatoria uniforme de los valores de los genes de los padres; es decir, se van eligiendo los genes de cada padre para el hijo, según el resultado de la comparación entre un número generado aleatoriamente y uno preestablecido (Syswerda 1989).

Otros operadores de cruzamiento para el resto de las representaciones de los cromosomas pueden encontrarse en (Herrera et al. 1998).

1.3.5.2 Operador de mutación

En este operador un gen es elegido aleatoriamente del cromosoma seleccionado y su valor es cambiado por otro que pertenezca al dominio de los genes. En el caso de la representación binaria de los cromosomas, este operador es muy sencillo de implementar

ya que basta con intercambiar "1" por "0" y viceversa (Holland 1975; Goldberg 1989). La Figura 1.5 Operador de mutaciónmuestra un ejemplo de aplicación del operador de mutación.

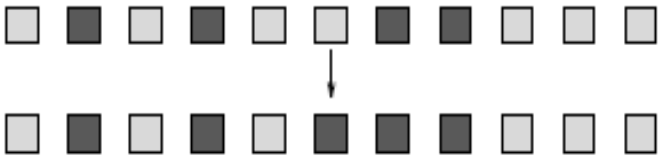

Figura 1.5 Operador de mutación

1.4 La Inteligencia Artificial para encontrar un mejor multiclasificador.

En varios campos de la Inteligencia Artificial proponen y ofrecen posibles soluciones para construir un multiclasificador que supere los resultados obtenidos por los clasificadores individuales.

En (Partalas et al. 2009) se aplica la perspectiva del aprendizaje reforzado (Reinforcement Learning) para reducir el número de clasificadores a usar en la construcción de un multiclasificador, usando para ello el algoritmo *Q-learning*. Los resultados obtenidos muestran que la propuesta de algoritmo que se presenta produce conjuntos de pequeño tamaño y por tanto reduce los recursos computacionales y de memoria que se necesitan. Al igual que en otros métodos basados en búsqueda de subconjuntos, el espacio de estados que se tienen que explorar crece exponencialmente con el número de clasificadores.

Los Algoritmos Genéticos también han sido usados en la búsqueda de un mejor multiclasificador. En (Kuncheva & Jain 2000) se sugieren dos formas de usar los AG para diseñar un sistema multiclasificador. La primera versión selecciona subconjuntos disjuntos de rasgos para ser usados por los clasificadores individuales, mientras que la segunda selecciona subconjuntos de rasgos en los que puede haber solapamiento y también el tipo de clasificadores a usar. El objetivo principal de la configuración de las dos versiones de AGs fue explorar el conjunto de soluciones, dado que la función de adaptabilidad era altamente multimodal. De las dos versiones la segunda demostró ser la mejor opción a usar, mientras que la primera se recomienda usarla en bases de datos en donde haya un número alto de rasgos dado que se puede inducir un mal desempeño de los clasificadores individuales si el número de rasgos usados es pequeño.

1.5 Consideraciones finales del capítulo

Los problemas vinculados a la multiclasificación abundan en la actualidad dado el gran número de información con que se cuenta en determinadas áreas de la ciencia. En este capítulo se explican algunos de los algoritmos existentes para construir un sistema multiclasificador, concepto de reciente utilización por los investigadores. Además, se muestran las medidas de diversidad existentes en la literatura para cuantificar la diversidad en un conjunto de clasificadores. Po último se describen los fundamentos teóricos principales de los Algoritmos Genéticos, meta heurística que será usada en la solución de nuestro problema.

DISEÑO E IMPLEMENTACIÓN SOBRE EL SOFTWARE DASDE

2 CAPÍTULO 2: DISEÑO E IMPLEMENTACIÓN SOBRE EL SOFTWARE DASDE

En este capítulo se presenta primeramente un breve resumen del diseño e implementación del software DASDE v1.0, también se describe la forma en que fue diseñada e implementada una variante de AG para resolver la problemática que se presenta en nuestro trabajo. Además, se presentan las modificaciones necesarias en el software anterior para incorporar la implementación de esta meta heurística.

Por último, se muestra el diagrama de las principales clases incorporadas al software obteniendo finalmente una versión más completa del mismo con mayores funcionalidades.

2.1 Diseño e Implementación del software DASDE v1.0.

La herramienta DASDE® 1.0 (Diversity Analysis, Selection and Discovery of best Ensemble from bases models) obtenida en (Meneses Gómez & Cedré Gutiérrez 2013) es un software desarrollado por investigadores de la Universidad Central "Marta Abreu de Las Villas" en el año 2013, con el lenguaje de programación Java, multiplataforma, y utiliza el software *weka* como biblioteca; está destinado a diversos tipos de usuario, como estudiantes, especialistas o investigadores en química, biología, computación o ramas similares.

Requerimientos del software

El software DASDE V1.0 necesita como mínimo:

1. Tener instalada una Máquina virtual de Java (Java(TM) Runtime Environment) igual o superior a la versión 7
2. 1 Gb de memoria RAM

Descripción del 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 .csvy .arff deben estar separadas por coma.

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 químicos, 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 modelos de 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).

Descripción de las ventanas del software

El software muestra una interfaz visual que contiene cuatro partes o paneles fundamentales dedicados a distintas funcionalidades.

En el primer panel de la aplicación se cargan los datos y se muestra información de los mismos (número de compuestos químicos, cantidad de clasificadores individuales, los valores del atributo de clasificación, etc.), además le permite seleccionar los casos con los que se realizarán los estudios (correspondientes a la serie de entrenamiento) y los casos con los que se comprobará los resultados (correspondientes a la serie de predicción), teniendo en cuenta que los valores de los atributos corresponde a la salida individual de cada modelos de clasificación, los que pueden ser: (i) la probabilidad de que el compuesto sea activo (P) o (ii) la diferencia de la probabilidad de que sea activo menos la probabilidad de que sea inactivo (ΔP : [Pactivo-Pinactivo]*100).

El segundo panel es un pre-procesamiento de los modelos individuales donde se obtienen estadísticas de cada uno de ellos. Este es el que permite ver el comportamiento de los modelos, mostrando su matriz de confusión y los parámetros estadísticos que indican la robustez de los clasificadores: Coeficiente de correlación de Matthews, Exactitud, Sensibilidad, Especificidad y Razón de Falsa Activos (RFA) para la clase activa; además permite seleccionar las variables con las que se desean realizar los cálculos en el próximo paso.

El tercer panel, corresponde al cálculo de las combinaciones de clasificadores más diversos, a través del uso de las medidas de diversidad pareadas y no pareadas, se recuerda que en este trabajo se incorporó la implementación de dos de ellas: Medida de desacuerdo entre expertos y Diversidad generalizada, una vez obtenidas estas combinaciones entonces en el siguiente panel solo se tienen en cuenta aquellos clasificadores seleccionados como diversos entre sí. En el último panel se obtienen las combinaciones de modelos que aumentan la mejor exactitud individual de los mismos, el usuario puede seleccionar varias reglas para combinar las salidas de los

clasificadores. Además, puede cambiar el valor de corte a partir del cual se define cada clase, o escoger la forma en que se desea visualizar los mejores modelos ensamblados.

2.2 Modificaciones sobre el software

En los problemas de quimio-bioinformática se aplican una gran cantidad de clasificadores individuales lo que hace difícil la elección de cuáles de ellos combinar, con tan solo un pequeño número de ellos se pueden generar grandes cantidades de combinaciones y por lo tanto es muy difícil buscar la mejor combinación donde el resultado que se obtenga supere el mejor obtenido por los clasificadores individuales, además es difícil también garantizar que los clasificadores tenidos en cuenta para combinar sean diversos entre sí, lo cual es muy importante porque como se mencionó en el capítulo anterior no tendría sentido combinar clasificadores idénticos.

En el caso de la herramienta computacional "DASDE®" el proceso de buscar los clasificadores más diversos entre sí y la mejor combinación que supere los resultados individuales de los clasificadores tarda bastante y en algunos casos es bastante difícil encontrarla, ya que se implementó una búsqueda exhaustiva, por lo que se hace necesario realizar varias modificaciones al software para incorporar la implementación de la meta-heurística Algoritmos Genéticos, la cual ofrece la posibilidad de no tener que explorar todo el espacio de búsqueda de las posibles soluciones.

Para incorporar las nuevas medidas de diversidad en el software *DASDE* se agregó la implementación de las mismas en el método *calcularMed()* de la clase *CalcDiversity*..

A continuación, se muestra cómo se agrega una nueva medida de diversidad no pareada

```
Switch (med) {
      Case <número>:
               Medida();
      Break;
            (...)
}
```

Donde cada caso del *switch* representa las diferentes medidas de diversidad, por lo que el número de la nueva medida debe ser consecutivo a los anteriores. En cada *case* se llama al método de la nueva medida de diversidad, guardando cada valor en el arreglo *result2*.

Para incorporar la meta heurística AG al sistema se diseñó e implementó un nuevo módulo constituido por un paquete llamado AG, donde se definen las características específicas del algoritmo genético.

Clases contenidas en el paquete:

- ✓ *Cromosoma*: Encargada de la construcción de los cromosomas y donde se obtienen los modelos más diversos de acuerdo a distintas medidas de diversidad, además contiene los métodos fundamentarles como **crearGen ()** y **calcularFunction ()** donde esta última es la función objetivo del algoritmo genético.
- ✓ *Cromosoma_Ensamble*: Encargada de la construcción de los cromosomas y donde se obtienen las combinaciones de los modelos que superan la mejor la mejor exactitud individual, además contiene los métodos fundamentarles como **crearGen ()** y **calcularFunction ()** donde esta última es la función objetivo del algoritmo genético.
- ✓ *Población*: Encargada de la construcción de la población, además contiene los métodos que utiliza el algoritmo genético en la construcción de las soluciones como el método **selection()** que es el encargado de selecciona los mejores individuos de la población y para ello aplica el método de la ruleta; el método de cruzamiento **crossover ()** que cruza los cromosomas de dos maneras con el cruzamiento clásico y el uniforme, el método de **mutation ()** donde realiza la variación genética del cromosoma, también se encuentra el método **startEvolution ()** que es el encargado de buscar las mejores soluciones posibles.
- ✓ *Diversidad:* se encarga de calcular el valor de una medida de diversidad para un individuo en particular. También contiene la implementación de las distintas medidas de diversidad.
- ✓ *Matriz _Diversidad:* almacena información referente a los clasificadores de una solución en particular, es la encargada de calcular el valor de la matriz *A, B, C, D* explicada en el epígrafe 1.2.1 y devolver el valor de la matriz correspondiente a cada par de clasificadores, se recuerda que solo se calcula dicha matriz cuando aplicamos medidas de diversidad pareadas.
- ✓ *Multiclasificador:* se encarga de obtener el multiclasificador, donde se utilizan varias reglas de combinación que pueden ser elegidas por el usuario. Además, permite calcular una serie de estadísticas como: Coeficiente de correlación de

Matthews, Exactitud, Sensibilidad, Especificidad, Razón de Falsa Activos (RFA), entre otras. Esta clase también contiene la implementación de las diferentes funciones de combinación.

- ✓ *Utiless:* contiene la implementación de un conjunto de métodos que son utilizados en diferentes clases.

A continuación, se muestran los diagramas de clases correspondiente al paquete AG explicado anteriormente y se detalla la relación existente entre las clases y entre otros paquetes.

En la Figura 2.6 Diagrama de Clasesse muestra el diagrama de clases y en la Figura 2.7 Diagrama de Paqueteel diagrama de paquete.

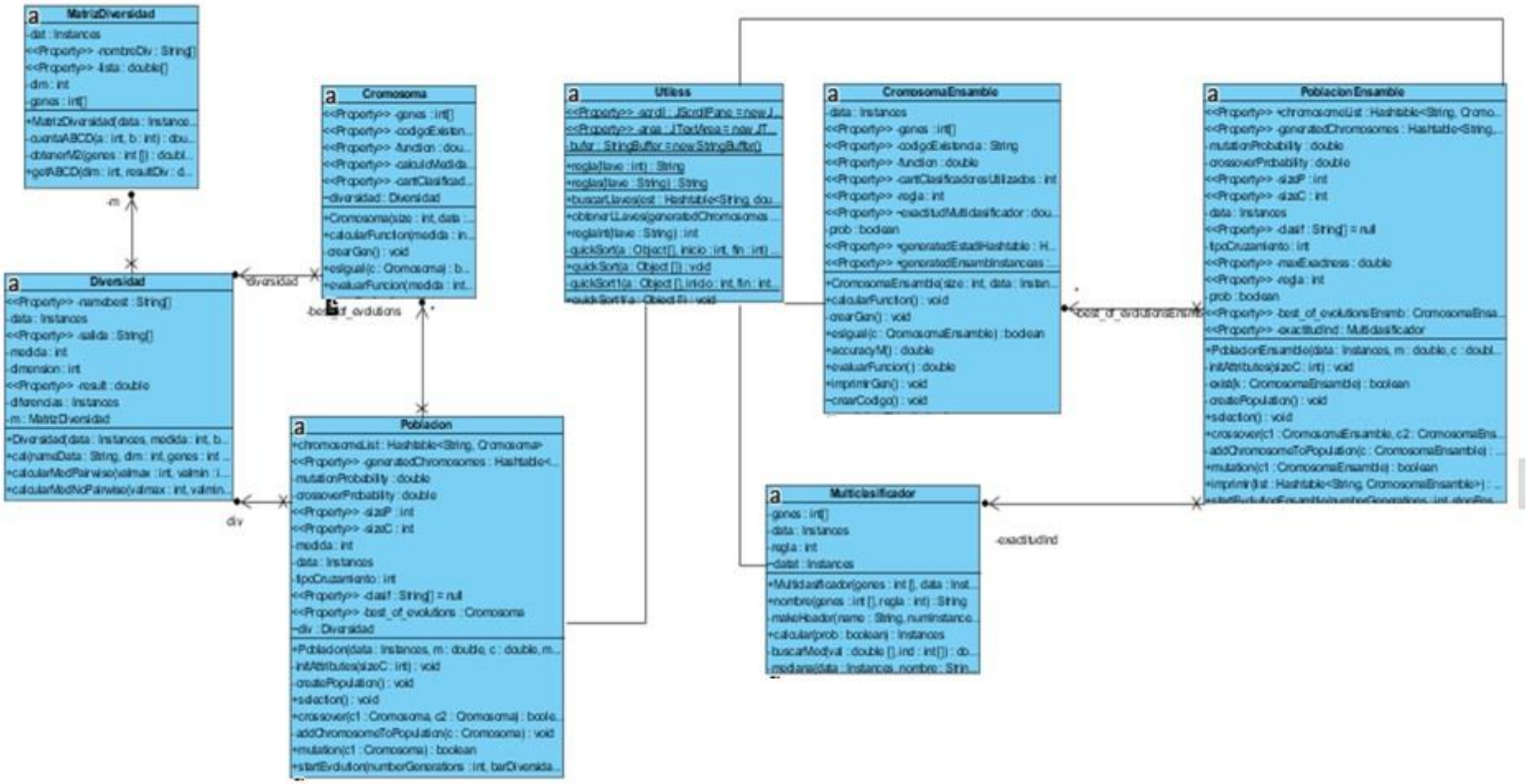

Figura 2.6 Diagrama de Clases

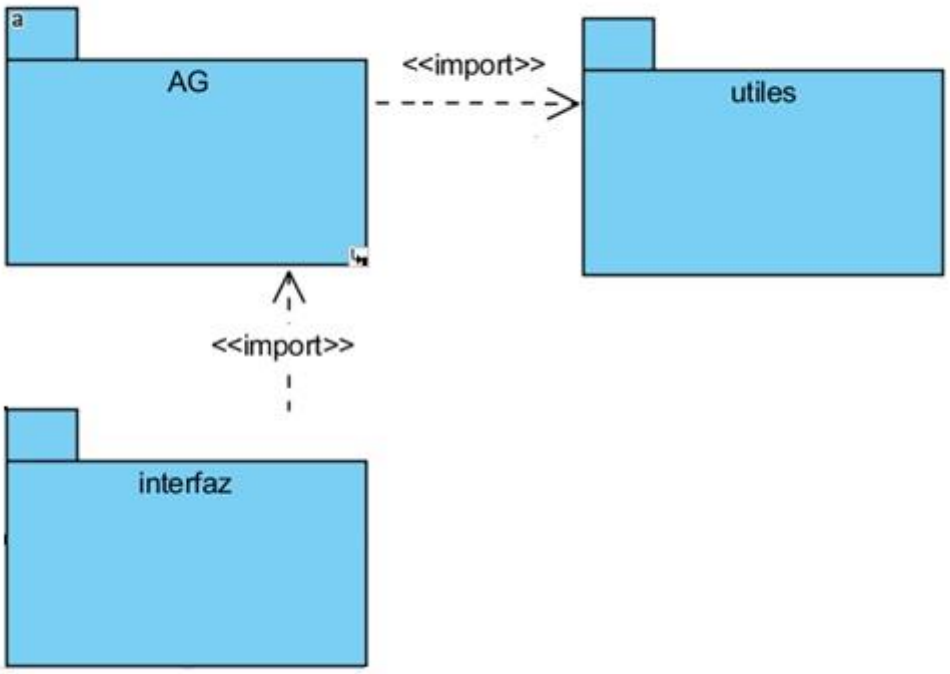

Figura 2.7 Diagrama de Paquete

2.2.1 Configuración del cromosoma

El cromosoma será el encargado de representar las posibles soluciones del problema a resolver. En este caso, se trata de decidir qué clasificadores incluir en la solución. Por ello, se decide utilizar la representación basada en cadenas de bit donde un cromosoma Cx es configurado de la siguiente forma:

$$C_x = (g_1, g_2, \dots, g_i) \; donde \; g_i = \begin{Bmatrix} 0 \; , si \; el \; clasificador \; i \; no \; está \; presente \\ 1 \; , si \; el \; clasificador \; i \; está \; presente \end{Bmatrix} \quad \text{Eq.1.17}$$

2.2.2 La población. Elementos principales

El análisis de la población incluye varios elementos. Existen varias investigaciones relativas a la influencia del tamaño de la población en la convergencia de un AG. Es lógico pensar que el trabajo con poblaciones pequeñas corre el riesgo de representar pobremente el espacio de soluciones. Por otra parte, las poblaciones de gran tamaño consumen mayor tiempo computacional. Sobre esta disyuntiva y como un trabajo teórico, Goldberg obtuvo en su investigación que el tamaño óptimo de una población de cadenas binarias, crece exponencialmente con la longitud de la cadena (Goldberg 1989). Sin embargo, en diferentes resultados empíricos muchos autores sugieren tamaños de poblaciones tan pequeños como 30 individuos. Debido a la gran cantidad de modelos de

clasificadores individuales que se encuentran en las bases de casos se decidió que para la variante de algoritmo que se propone en este trabajo utilizar el tamaño de la población igual a $L * 2$, donde L es la cantidad de clasificadores; o sea, el tamaño del cromosoma.

La población inicial fue generada aleatoria. Cada cromosoma es generado aleatoriamente, tomando cada gen valor 0 o 1 en dependencia de un número r generado aleatoriamente; si r es mayor que 0.5 se incluye el clasificador representado por el i-ésimo gen; en caso contrario, no se incluye.

Cada iteración simple del AG se inicia con una población con la cantidad especificada anteriormente. Los nuevos cromosomas generados a partir de los operadores genéticos de recombinación serán agregados a la población. Finalmente, a la población se eliminarán aquellos cromosomas que probabilísticamente son los de menor valor en la función objetivo, hasta quedarse con el tamaño establecido. Dado que el algoritmo fue pensado para aplicar con cualquier cantidad de clasificadores, se restringió la posibilidad de que en la población existiesen cromosomas repetidos.

2.2.3 Operadores utilizados para la formación de nuevas generaciones

Una vez generar la población inicial se crea una población intermedia mediante la selección, y sobre esta población se aplicarán operadores para obtener nuevos individuos y con mejor adaptabilidad expresada en términos de la función objetivo. Cada individuo es seleccionado, en dependencia de su calidad, para competir en el proceso evolutivo. A partir de estos cromosomas seleccionados, son creados nuevos cromosomas por medio del cruzamiento y la mutación, pudiendo ser incluidos o no en la nueva población. En los subepígrafe siguientes se explican los operadores utilizados para simular la recombinación genética y el mecanismo de selección natural.

2.2.3.1 Operador de cruzamiento

La obtención de nuevos cromosomas a partir de los existentes es una de las características más interesantes e importantes en el procedimiento de los AG. De ahí que la formulación depende en gran parte de los requisitos del problema y de la representación que se haya realizado (Morales 2014).

En el caso del operador de cruzamiento, se seleccionan fragmentos del genotipo de cromosomas que individualmente no soy muy buenos pero una vez mezclarlos pueden resultar en una solución mejor que la que se tenía. Según la literatura consultada y revisada existen varias formas de definir este operador. En el presente trabajo se decidió implementar el operador clásico de cruzamiento en un punto y el cruzamiento uniforme.

En el cruzamiento en un punto, se seleccionan manera aleatoria de la población intermedia los dos cromosomas que van a actuar como padres. También de forma aleatoria es elegido el locus[1] a partir del cual se va a cruzar. Como resultado de aplicar este operador de cruzamiento se obtienen dos nuevos cromosomas. La Figura 1.4 Cruzamiento clásico en un puntomuestra detalladamente este proceso. En el caso del cruzamiento uniforme, cada padre tiene igual probabilidad de aportar sus genes al único individuo resultante de aplicar este operador. Si un número generado aleatoriamente es menor o igual que 0.5 el gen será tomado del primer padre; en caso contrario, será tomado del segundo (Morales 2014). La Figura 2.8 Cruzamiento Uniforme muestra la forma en que funciona este operador.

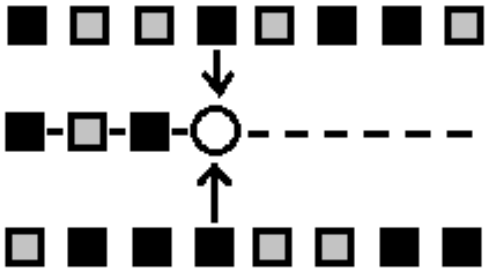

Figura 2.8 Cruzamiento Uniforme

Si cuando concluya el proceso de recombinación genética por medio del cruzamiento, los nuevos cromosomas ya existían en la población, entonces se procede a forzar que ocurra una mutación para obtener cromosomas nuevos y diferentes.

2.2.3.2 Operador de mutación.

La implementación de este operador es muy sencilla y trasparente debido a que solamente se tiene que intercambiar el alelo[2] del gen que se va a mutar. De manera que, si en un cromosoma el quinto gen indica que ese clasificador está presente en dicha combinación, luego de aplicar la mutación en ese locus resultaría que ese clasificador ya no se incluye en la combinación. La Figura 1.5 Operador de mutaciónejemplifica este operador.

Operador de selección natural

El proceso se forma una población intermedia de cromosomas sobre los cuales se aplicarán los operadores anteriores para obtener una nueva población, con cromosomas de más calidad que los existentes. Para la selección de los cromosomas que formarán

[1]Posición que ocupa un gen dentro del cromosoma.
[2] Valor asociado a un gen.

parte de esta población intermedia se usa la función objetivo, la que evaluada en cada uno de los individuos determinará su selección para participar en la recombinación genética.

Una vez formada la población intermedia, se aplican los operadores de mutación y cruzamiento atendiendo a la probabilidad de que puedan ocurrir o no, y los cromosomas resultantes se agregan a la población original. Teniendo en cuenta las características del problema que se quiere solucionar el proceso de selección se divide o ejecuta en dos momentos; antes de efectuar la recombinación y después de haberla aplicado. La primera se hace con el objetivo de que solo los mejores individuos sean los que se reproduzcan y la segunda es debida a que los individuos nuevos, dentro de la población original, pueden o no tener las capacidades adaptativas para sobrevivir a la próxima generación y, por tanto, algunos de ellos son desechados.

En la primera selección se toman los cromosomas de mayor valor en la función objetivo de la población inicial y en la segunda selección se aplica el método de la ruleta sobre la población aumentada con los cromosomas resultantes de aplicar la recombinación genética; de forma que se reduzca su tamaño al tamaño especificado en el sub-epígrafe 2.2.2, no permitiéndose en este caso la selección de un mismo cromosoma más de una vez. A continuación, se muestra el seudocódigo que realiza el proceso de la segunda selección; es decir, el método de la ruleta.

P1: Inicializar variables

$\boldsymbol{total} \leftarrow 0,\ \boldsymbol{ruleta}[\,],\ \boldsymbol{tempP} \leftarrow \emptyset$

P2: Obtener los valores de $f(C_X)$ de los cromosomas de la población P' formada de adicionar los nuevos cromosomas resultantes de la recombinación a la población P.

$\boldsymbol{funciones}[\,] \leftarrow \boldsymbol{obtenerValoresFunciones}();$

$\boldsymbol{total} \leftarrow \sum \boldsymbol{funciones}[\boldsymbol{C_i}]$

P3: Construir la ruleta:

$$\boldsymbol{ruleta}[\mathbf{1}] \leftarrow \frac{\boldsymbol{funciones}[\mathbf{1}]}{\boldsymbol{total}}$$

Para $\boldsymbol{i} = \mathbf{2}, \mathbf{3}, \ldots, |\boldsymbol{P'}|$

$$\boldsymbol{ruleta}[\boldsymbol{i}] \leftarrow \boldsymbol{ruleta}[\boldsymbol{i} - \mathbf{1}] + \frac{\boldsymbol{funciones}[\boldsymbol{i}]}{\boldsymbol{total}}$$

P4: Construcción de la población de la siguiente generación:

Mientras $|\,\boldsymbol{tempP}\,| \neq |\boldsymbol{P}|$

$\boldsymbol{posAleatoria} \leftarrow \boldsymbol{obtener\_posicion}(\boldsymbol{randomValue})$

$\boldsymbol{value} \leftarrow \boldsymbol{ruleta}[\boldsymbol{posAleatoria}]$

$\boldsymbol{tempP} \leftarrow \boldsymbol{tempP} \cup \boldsymbol{value}$

$\boldsymbol{ruleta} \leftarrow \boldsymbol{ruleta} / \boldsymbol{value}$

$actualizar\_ruleta()$

P5: Actualizar la población de la generación siguiente:

$\boldsymbol{P} \leftarrow \boldsymbol{tempP}$

Los procedimientos *obtener_ posicion() y actualizar_ruleta()* son utilizados, el primero para obtener la posición del cromosoma que se va a seleccionar para la nueva población y el segundo para recalcular los intervalos de probabilidades de cada cromosoma a partir de uno retirado de la selección.

2.2.4 Restricciones de problema

En la modelación del problema se tiene en cuenta varias restricciones, a continuación, se muestran cada una de ellas:

1. $2 \leq |x| \leq 10$

En el proceso de recombinación genética para obtener nuevos individuos se desechan aquellos con una cantidad de clasificadores mayores a 10, ya que es una restricción del cliente final del software.

2.2.5 Implementación de Algoritmo genético con Medidas de diversidad.

Como se describe en el capítulo uno, la configuración del AG depende mucho del tipo de problema al que se le va a dar solución. Donde el problema planteado se ajusta a los requerimientos del algoritmo. Una vez definida la configuración y la representación de

cada uno de los elementos necesarios, se procede a establecer los operadores genéticos que serán los encargados de que cada población evolucione, y en cuyo proceso debe encontrase la solución esperada. Estos operadores deben responder a las restricciones del problema y, por tanto, deben ser en ocasiones adaptados. En este epígrafe se hará referencia a la variante de AG que fue implementada para incorporarla en el panel de diversidad donde el mismo se encarga de encontrar las combinaciones de modelos más diversos.

2.2.5.1 Descripción de la función objetivo.

En nuestro problema, se quiere obtener la mayor diversidad entre los clasificadores que se combinen en el sistema. Por ello, la función que determina la calidad de un cromosoma x se define como:

$$f(C_x) = Diversidad(C_x) \qquad \text{Eq.1.18}$$

Donde la diversidad del cromosoma es el valor resultante de aplicar una medida de diversidad (o un conjunto de ellas) a los clasificadores de C_x. Por tanto, la función objetivo a través del proceso evolutivo será:

$$\max_{0 \le x \le P} f(C_x) \qquad \text{Eq.1.19}$$

donde P es el tamaño de la población.

Al finalizar el proceso evolutivo de la población, el individuo (s) (o cromosoma (s)) de mayor valor en la función objetivo debe ser portador de la mejor combinación que se está buscando respecto a la diversidad del conjunto de clasificadores.

2.2.6 Implementación de Algoritmo Genético para obtener las mejores combinaciones individuales.

En este epígrafe se hace referencia a la variante de AG que fue implementada para incorporar en el último panel la búsqueda de combinaciones que superen la mejor exactitud de los modelos individuales.

2.2.6.1 Modelos individuales incluidos en conjunto de partida.

A medida que se obtengan resultados en el panel de diversidad, se seleccionan automáticamente los modelos incluidos en las combinaciones más diversas para formar parte del conjunto inicial de modelos que serán combinados buscando las soluciones con mejor exactitud.

2.2.6.2 Descripción de la función objetivo

En esta variante de la meta heurística se quiere obtener la mayor exactitud entre los clasificadores que se combinen en el sistema y a la vez las soluciones con menor

cantidad de clasificadores incluidos. Por ello, la función que determina la calidad de un cromosoma x se define como:

$$f(C_x) = Exactitud(C_x) + \left(1 - \frac{x}{n}\right) \quad \text{Eq.1.20}$$

Donde x es la cantidad de clasificadores incluidos en la solución y n la cantidad máxima posible de clasificadores en la solución.

La exactitud del cromosoma se obtiene combinando las salidas de los clasificadores a través de determinadas reglas como: Voto mayoritario, Promedio o Average, Máximo, Mínimo, Mediana y Producto. la función objetivo a través del proceso evolutivo será:

$$\max_{0 \le x \le P} f(C_x) \quad \text{Eq.1.21}$$

donde P es el tamaño de la población.

Supuestamente, al finalizar el proceso evolutivo de la población, el individuo (o cromosoma) de mayor valor en la función objetivo deberá ser portador en sus genes de la mejor combinación que se está buscando.

2.3 Conclusiones parciales del capítulo.

En este capítulo se describe la forma en que fue diseñada e implementada una variante de AG para resolver la problemática que se presenta en nuestro trabajo. Además, se presentan las modificaciones necesarias en la versión anterior del software para incorporar la implementación de esta meta heurística.

Por último, se muestra el diagrama de las principales clases incorporadas al software obteniendo finalmente una versión más completa del mismo con mayores funcionalidades. Por último, se muestra el diagrama de las principales clases incorporadas al software obteniendo finalmente una versión más completa del mismo con mayores funcionalidades.

DISEÑO DE EXPERIMENTOS Y ANÁLISIS DE SUS RESULTADOS

3 CAPÍTULO 3 DISEÑO DE EXPERIMENTOS Y ANÁLISIS DE SUS RESULTADOS

En este capítulo se muestran aplicaciones con bases de datos reales para valorar el nivel de satisfacción de los investigadores del Centro de Bioactivos Químicos (CBQ) de nuestra universidad ya que la herramienta surge como una necesidad de sus investigaciones. Luego se muestra una comparación con los resultados obtenidos en la primera versión del software.

3.1 Descripción general de los experimentos

Para la validación de la nueva implementación se utilizaron dos bases de casos de modelos *QSAR* (*Quantitive Structure Activity Relationship*). La primera fue obtenida como parte de una investigación de tesis doctoral del Centro de Bioactivos Químicos que identifica compuestos antimaláricos (Machado Tugores 2013). La segunda formó parte de un proyecto de tesis de grado (Cisneros Matos 2007), que determina compuestos activos frente a procesos inflamatorios.

Para el análisis de ambas bases de casos se ejecuta el software utilizando la meta heurística implementada en los paneles (*Diversity Measure)* y (*Ensemble Methods),* comprobándose en las combinaciones obtenidas su robustez, según distintos parámetros estadísticos, en especial la exactitud y la razón de falsos activos (RFA).

En las ejecuciones se varió el número de generaciones entre 50 y 200, obteniéndose los mejores resultados con la configuración de 100 generaciones, además se utiliza indistintamente los operadores de cruzamiento clásico y uniforme, y las probabilidades de ocurrencia de una mutación y un cruzamiento de 0.25 y 0.75 respectivamente, los cuales fueron sugeridos por los resultados alcanzados en (Morales 2014).

3.2 Análisis de la base de datos Malaria

La Malaria es una enfermedad protozoaria de alto impacto social, provocando la muerte entre 1-3 millones de personas y entre 300-500 millones de afecciones anualmente. Está considerada como una de las enfermedades parasitarias tropicales más importantes del mundo. 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"*.(Machado Tugores 2013)

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 casos de 2314 compuestos, quedando dividida en dos conjuntos de 851 y 1.463 compuestos activos e inactivos respectivamente, utilizando un análisis de *clúster* de *k*-vecinos más próximos, implementado en el paquete estadístico *STATISTICA 6.0*. (Machado Tugores 2013), quedan divididos estos grupos en dos subconjuntos separados, la serie de entrenamiento (SE) con 638 activos y 1.097 inactivos y la serie de predicción (SP) con 213 activos y 366 inactivos. De esta forma se asegura que cualquier clase química (determinada por *clúster*) estará representada en ambas series de compuestos. Todos los compuestos fueron calculados usando el descriptor molecular *TOMOCOMD CARDD (TOpological Molecular COMputer Design Computer Aided Rational Drug Design*) (Marrero-Ponce et al. 2005; Meneses-Marcel et al. 2005; Marrero-Ponce et al. 2006; Marrero-Ponce et al. 2008; Meneses-Marcel et al. 2008; Marrero-Ponce et al. 2009). En total fueron obtenidos 37 modelos por Análisis Discriminante Lineal usando los índices lineales, bilineales y cuadráticos, basados en las relaciones de átomos. Teniendo en cuenta que el mejor modelo individual es el 37 con una exactitud de un 91.41% y una RFA de 8.75%.

Resultados y discusión

Se utilizaron los 37 modelos QSAR para la identificación de compuestos antimaláricos en la construcción del fichero de entrada. La diferencia de las probabilidades (ΔP) de las salidas individuales de cada modelo (variable "*y"* o probabilidad de que pertenezca a una clase o a la otra) fue utilizada como variable de entrada al software DASDE. La SE consta de 1735 casos, de los cuales 638 son activos y los 1.097 restantes inactivos. SP cuenta con 579 compuestos, de ellos 213 inactivos y 366 activos.

A los 37 clasificadores individuales introducidos en el software se les calcularon las siguientes medidas de diversidad: Desacuerdo (D), Doble Falta (DF), Coeficiente de Correlación, Q Estadístico, Varianza de Kohavi-Wolpert, Dificultad, Desacuerdo entre expertos y Diversidad Generalizada, se obtuvieron las combinaciones más diversas lográndose la siguiente selección de modelos (360 en total):

1,2,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,22,23,24,25,26,28,30,32,33,34,35,37.

En la Figura 3.9 Valores estadísticos de los modelos más diversos. se puede apreciar que los modelos presentan un buen comportamiento en la predicción de la actividad debido a que los valores de la exactitud (Acc por sus siglas en inglés) oscilan entre 80 y 90 aproximadamente y la razón de falsos activos (FPR por sus siglas en inglés) oscila entre 5 y 15, lo que es de gran interés para los investigadores.

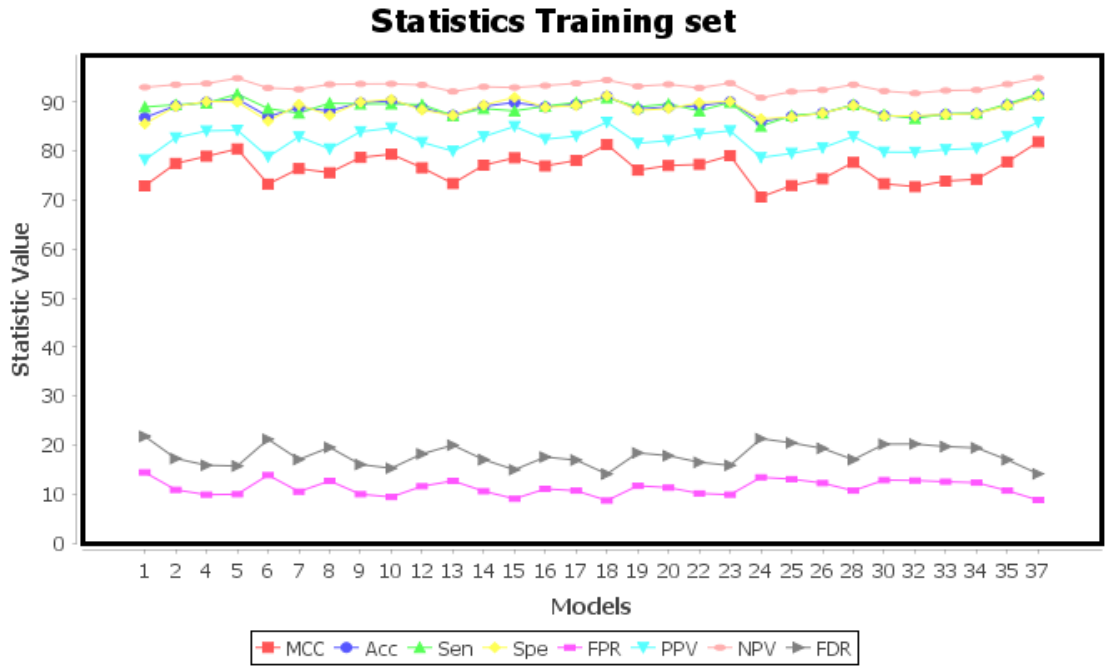

Figura 3.9 Valores estadísticos de los modelos más diversos.

A la selección anterior de modelos se le aplican las diferentes reglas de combinación para las salidas de los clasificadores y se obtienen, a través de la meta heurística, las combinaciones de clasificadores que superan la mejor exactitud individual, en la Tabla 3.3 Resultados de las mejores combinaciones que superan la exactitud individualse muestra la relación de los mejores modelos combinados por cada regla y las estadísticas correspondientes a cada uno, señalando aquellos que presentan una exactitud mayor que 91.41% y RFA menor que 8.75%. El mejor resultado se obtuvo en la selección correspondiente a las combinaciones de los modelos **4_5_17_35_37** según la regla Mínimo.

Modelos	Coeficiente de Matthews	Exactitud	Sensibilidad	Especificidad	RFA
2_15_18_19_22_35_37_V	81.88	**91.41**	91.54	91.34	**8.66**

9_12_15_16_18_A	75.85	87.84	93.73	84.41	15.59
4_10_18_X	78.76	89.68	92.32	88.15	11.85
4_5_17_35_37_N	**82.04**	**91.70**	**91.73**	**96.44**	**3.56**
5_18_19_20_37_M	81.73	91.3	92.01	90.88	9.12

Tabla 3.3 Resultados de las mejores combinaciones que superan la exactitud individual

En la Figura 3.10 Relación entre la exactitud de las mejores combinaciones obtenidas y RFA. se puede observar el comportamiento de la exactitud y la RFA de los mejores modelos ensamblados por cada una de las reglas. La combinación **4_5_17_35_37_N** presenta la mayor exactitud y la menor FRA, superando el mejor modelo individual.

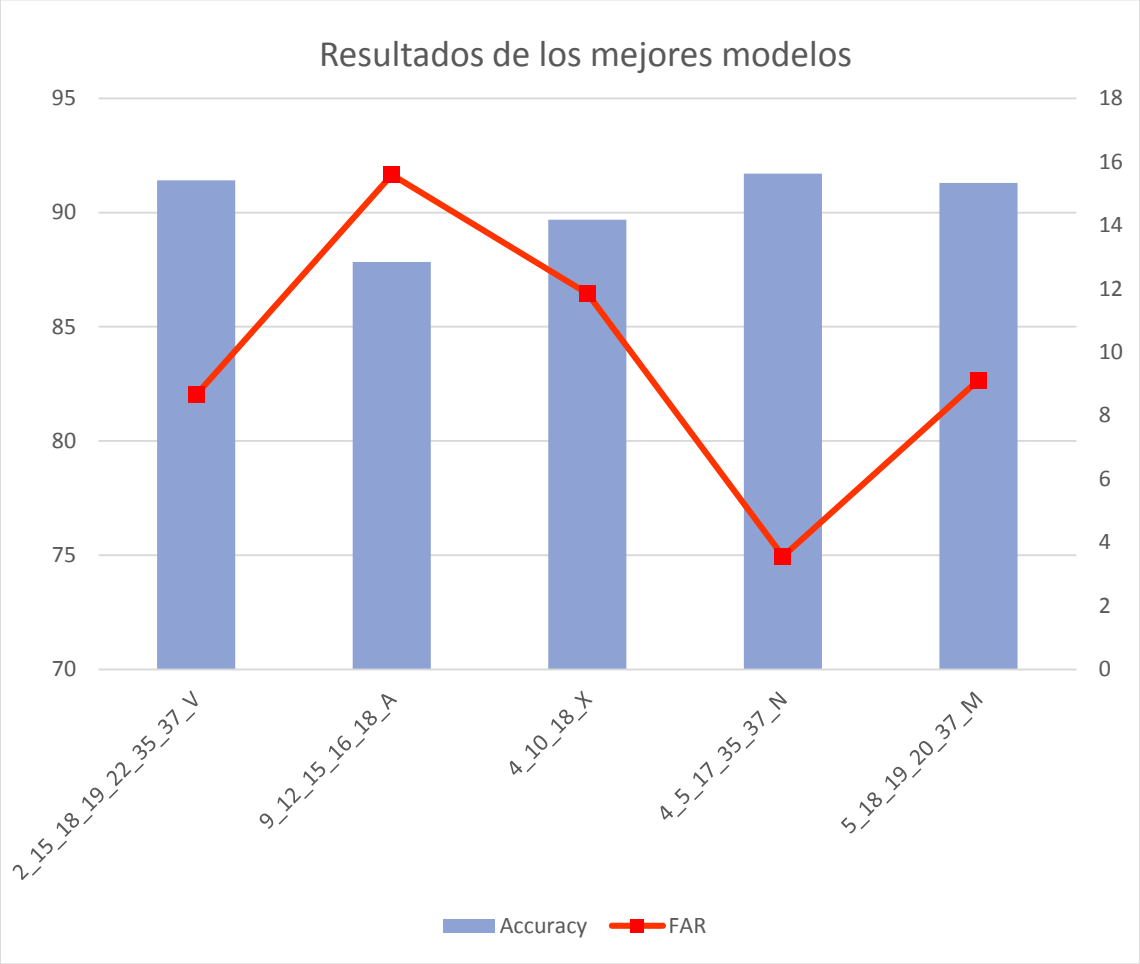

Figura 3.10 Relación entre la exactitud de las mejores combinaciones obtenidas y RFA.

3.3 Análisis de la base de casos Antinflamatoria

Los procesos inflamatorios se desarrollan como parte de los mecanismos de defensa que tienen lugar en el organismo en respuesta a estímulos de diversa naturaleza. Estos estímulos pueden abarcar desde lesiones provocadas por traumatismos mecánicos, procesos isquémicos, efecto de agentes patógenos como virus, parásitos o bacterias,

interacciones antígeno-anticuerpo desarrollados por la actividad de los diferentes componentes del sistema inmunológico, hasta la acción de diversos productos químicos. Solo en los casos en que estos factores puedan ser eliminados por la activación de los sistemas de defensa del organismo, no se presentan visiblemente las consecuencias patológicas del proceso inflamatorio. Pero si ellos se encuentran en grandes cantidades, presentes en sitios inusuales, o son capaces de modificar algunas de las funciones del sistema inmunológico, pueden ocurrir daños tisulares como consecuencia de las reacciones inmunes y esto a su vez facilitar el desarrollo de diferentes enfermedades inflamatorias (Cisneros Matos 2007)

La base de casos está compuesta por un total de 1213 moléculas (587 activas y 626 inactivas) recopiladas en (Cisneros Matos 2007). Mediante análisis de *clúster*, la base de casos quedó dividida en series (SE y SP), dichas están constituidas por moléculas activas e inactivas. Donde la serie de entrenamiento (SE) consta de 919 casos, de los cuales 150 son activos y los 476 restantes inactivos y la serie de predicción (SP) cuenta con 294 compuestos, de ellos 144 activos y los 443 inactivos. Teniendo en cuenta que el mejor modelo es 19 con una exactitud de un 88.68% y una RFA de 9.45%.

Resultados y discusión

A los 44 clasificadores individuales introducidos en el software se le calcularon las medidas de diversidad: Desacuerdo (D), Doble Falta (DF), Coeficiente de Correlación, Q estadístico, Varianza de Kohavi-Wolper, Medida de Dificultad, Desacuerdo entre expertos y Diversidad Generalizada obteniéndose la siguiente selección de clasificadores diversos (30 en total):

1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,19,20,21,22,23,24,25,29,31,35,37,39,41,42.

En la Figura 3.11 Valores estadísticos de los modelos más diversosse puede apreciar que los modelos presentan un buen comportamiento en la predicción de la actividad debido a que los valores de la exactitud (Acc por sus siglas en inglés) oscilan entre 80 y 90 aproximadamente y la razón de falsos activos (FPR por sus siglas en inglés) oscila entre 5 y 15, lo que es de gran interés para los investigadores.

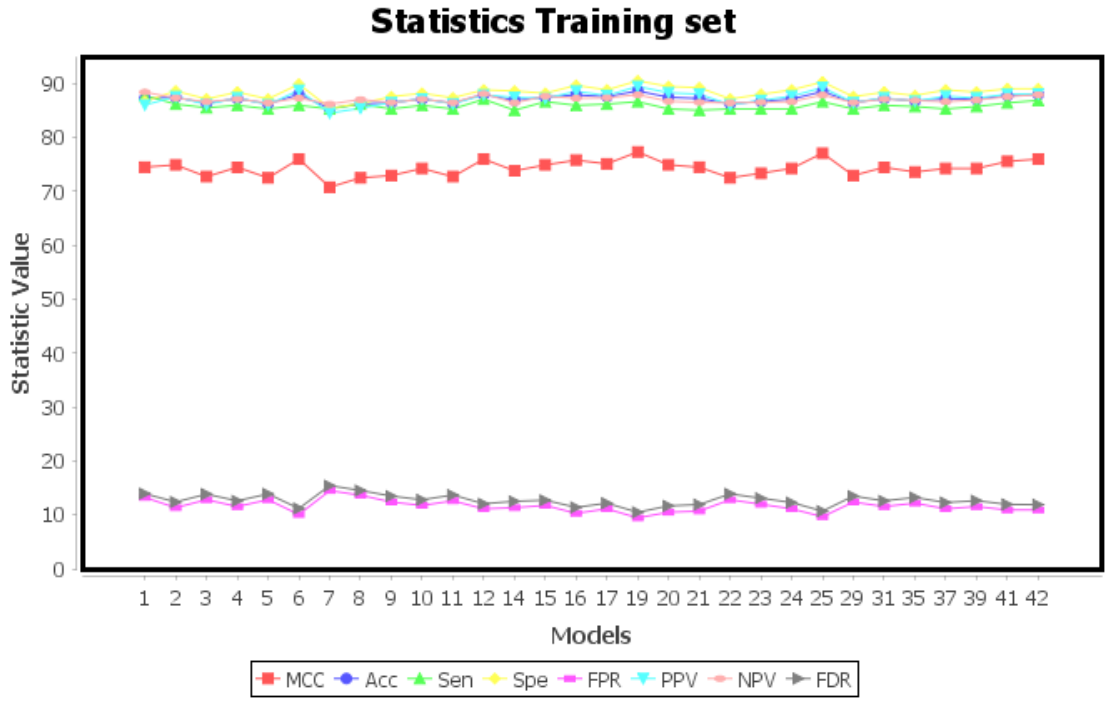

Figura 3.11 Valores estadísticos de los modelos más diversos

A la selección anterior de clasificadores diversos se le aplican las diferentes reglas de combinación para sus salidas. En la Tabla 3.**3** Resultados de las mejores combinaciones que superan la exactitud individualse muestra la relación de los mejores modelos combinados y las estadísticas correspondientes a cada uno, señalando aquellos que presentan una exactitud mayor que 88.68 % y RFA menor que 9.45 %. El mejor resultado se obtuvo en la primera selección correspondiente a la combinación de los modelos **2_8_11_16_21_22_26_28** según la regla de mínimo.

Modelos	Coeficiente de Matthews	Exactitud	Sensibilidad	Especificidad	RFA
2_4_6_9_18_23_30_V	78.44	89.23	87.13	91.18	8.82
12_15_21_25_A	71.23	85.42	89.84	81.30	18.7
6_17_20_X	72.12	85.96	88.94	83.19	16.81
2_8_11_16_21_22_26_28_N	**80.16**	**89.66**	**81.26**	**97.48**	**2.52**
3_6_11_14_22_23_27_28_30_M	78.01	89.01	86.91	90.97	9.03

Tabla 3.4 Resultados de Ensamble para los modelos más diversos

En la Figura 3.12 Valores de Exactitud y RFA de las combinaciones de los modelos se puede observar el comportamiento de la exactitud y la RFA de los mejores modelos ensamblado por cada una de las reglas. La combinación **2_8_11_16_21_22_26_28_N** presenta la mayor exactitud y la menor RFA, superando el mejor modelo individual

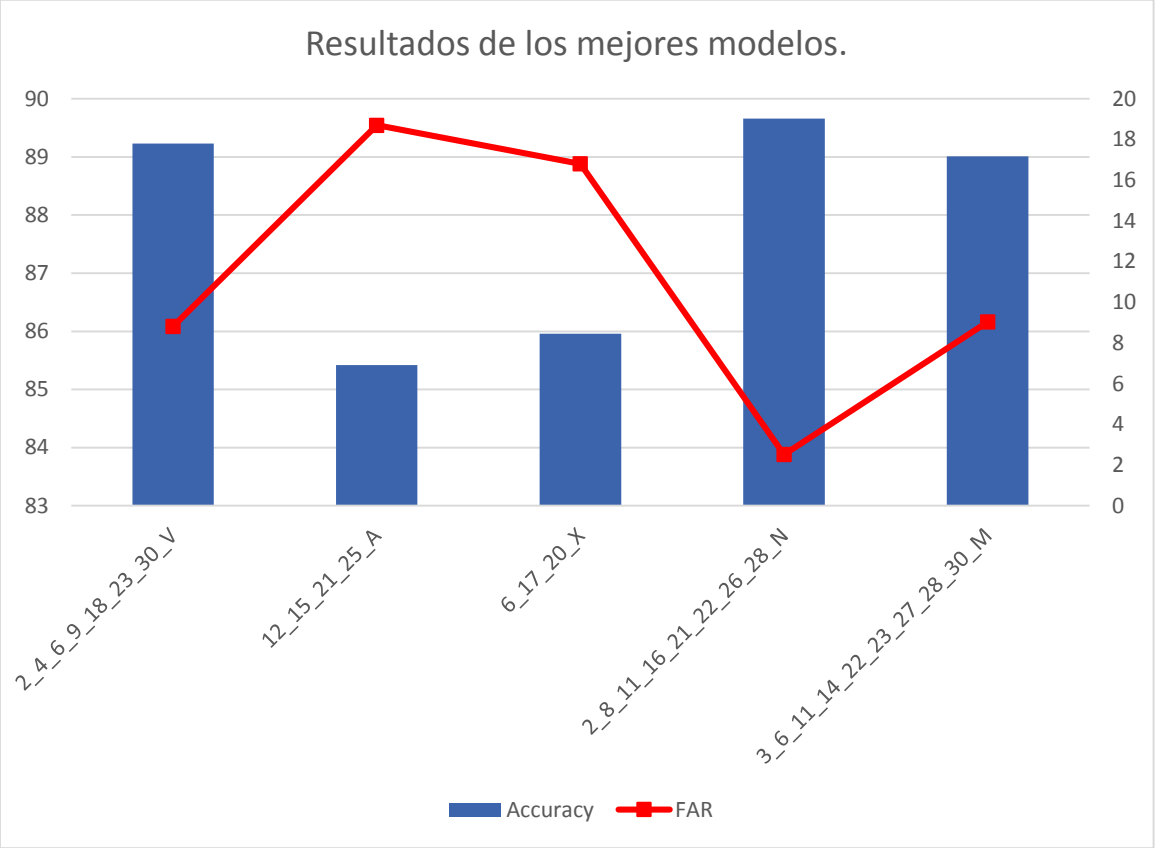

Figura 3.12 Valores de Exactitud y RFA de las combinaciones de los modelos

3.4 Comparación de los resultados con DASDE v1.0

En este trabajo estamos ante un problema donde no se garantiza encontrar la mejor solución posible en un tiempo razonable. Con el objetivo de encontrar buenas soluciones se decidió utilizar un método heurístico en contraposición a los *métodos exactos* que proporcionan una solución óptima del problema. Este *método heurístico* se limita a proporcionar una buena solución no necesariamente óptima, lógicamente, el tiempo invertido por un método exacto para encontrar la solución óptima de un problema difícil es de un orden de magnitud muy superior al del heurístico (pudiendo llegar a ser tan grande en muchos casos, que sea inaplicable).

Para garantizar que se obtuvieran buenas soluciones en tiempo razonable al aplicar la meta heurística AG, en este software se realizó un estudio en una computadora que presenta las siguientes propiedades: un procesador de Intel(R) Core(TM) i3-4169 CUP 3.60GHz, una memoria (RAM) 4,00 GB y un sistema operativo a 64-bit.

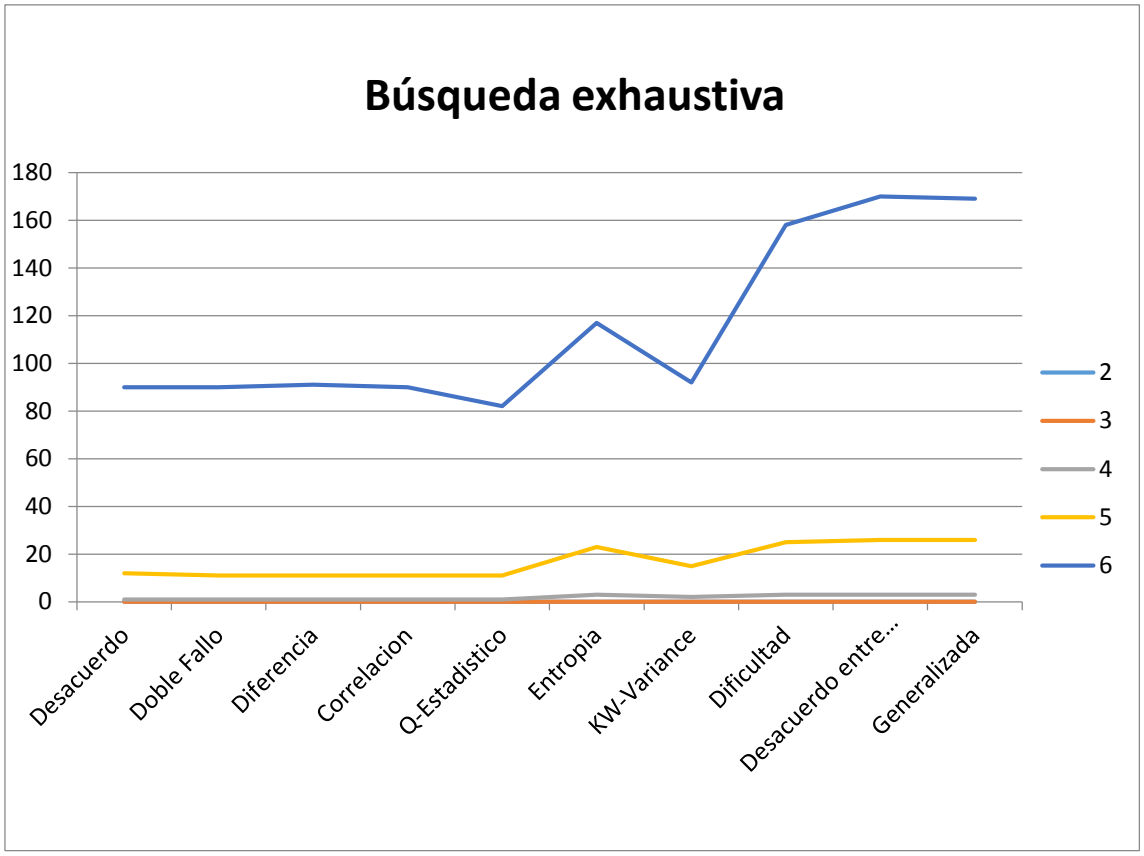

Figura 3.13 Tiempo de ejecución con búsqueda exhaustiva en el cálculo de combinaciones de clasificadores diversos con la base antimalárica

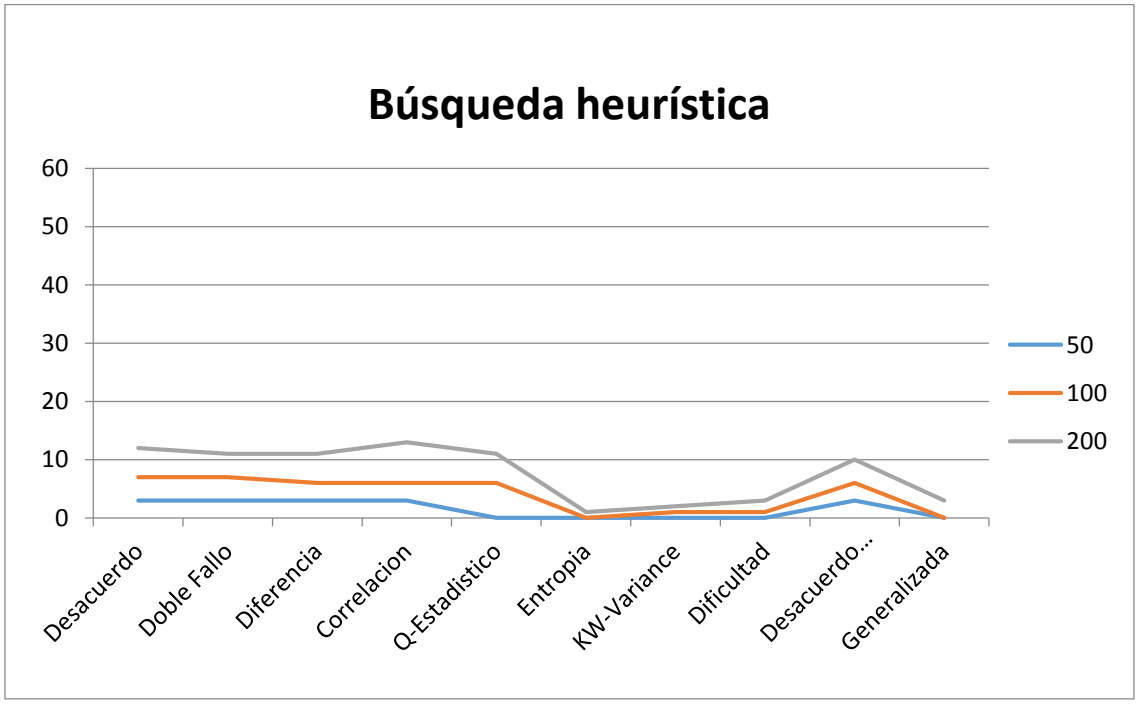

Figura 3.14 Tiempo de ejecución con búsqueda heurística en el cálculo de combinaciones de clasificadores diversos con la base antimalárica

En las Figura 3.13 y Figura 3.15 Tiempo de ejecución con búsqueda exhaustiva en el cálculo de combinaciones de clasificadores diversos con la base se puede observar para la base Malaria el comportamiento del tiempo de ejecución para encontrar las combinaciones de clasificadores más diversos calculando cada una de las medidas de diversidad usando búsqueda exhaustiva y heurística respectivamente.

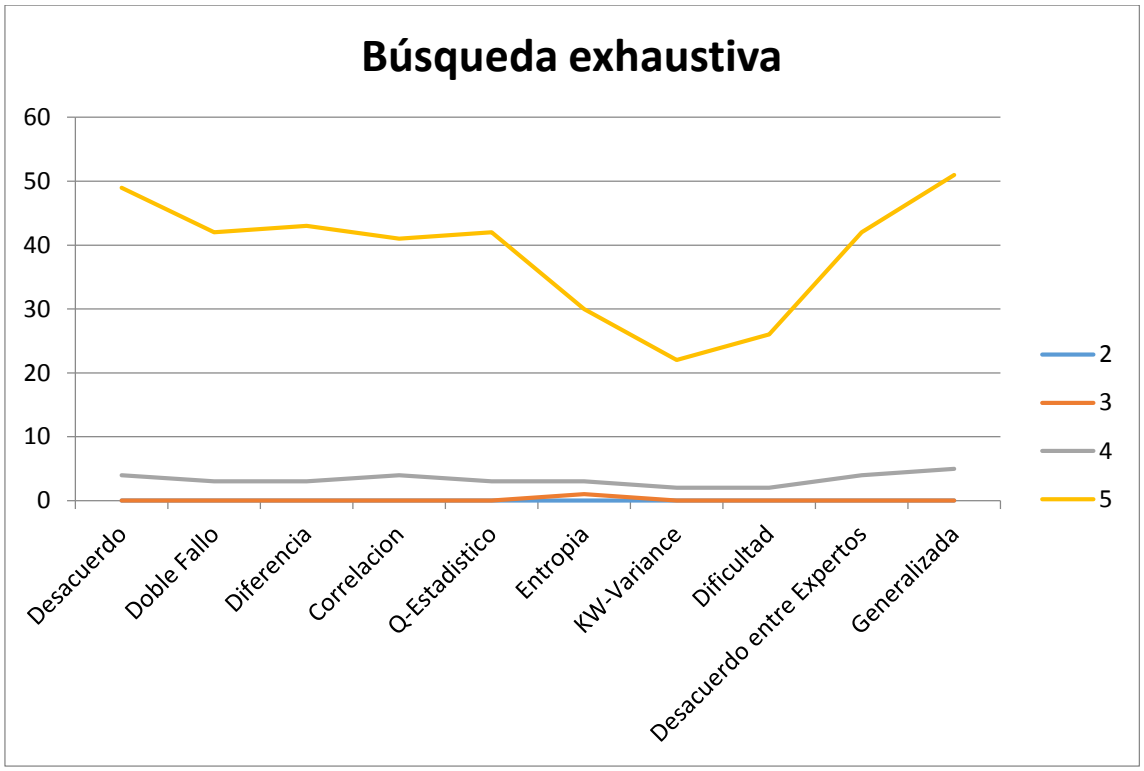

Figura 3.15 Tiempo de ejecución con búsqueda exhaustiva en el cálculo de combinaciones de clasificadores diversos con la base antinflamatoria

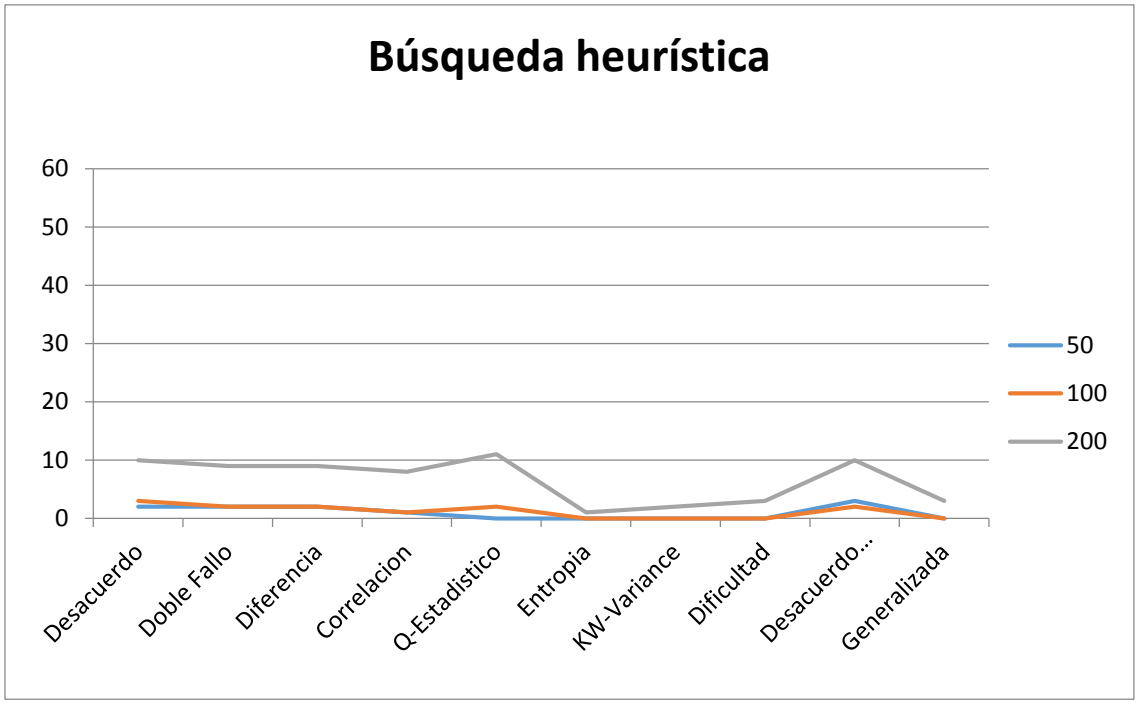

Figura 3.16 Tiempo de ejecución con búsqueda heurística con base antinflamatoria

En las Figura 3.15 Tiempo de ejecución con búsqueda exhaustiva en el cálculo de combinaciones de clasificadores diversos con la base antinflamatoriay en la Figura 3.16 Tiempo de ejecución con búsqueda heurística con base antinflamatoriase puede observar para la base antinflamatoria el comportamiento del tiempo de ejecución para encontrar las combinaciones de clasificadores más diversos calculando cada una de las medidas de diversidad usando búsqueda exhaustiva y heurística respectivamente.

En las Figura 3.17 Tiempo de ejecución en el panel *Diversity Measures* y Figura 3.18 Tiempo de ejecución en el panel *Ensamble Methods* se muestra una comparación relacionada con el tiempo de ejecución de ambas búsquedas: exhaustiva y heurística, específicamente en la base de datos *Malaria* analizada anteriormente.

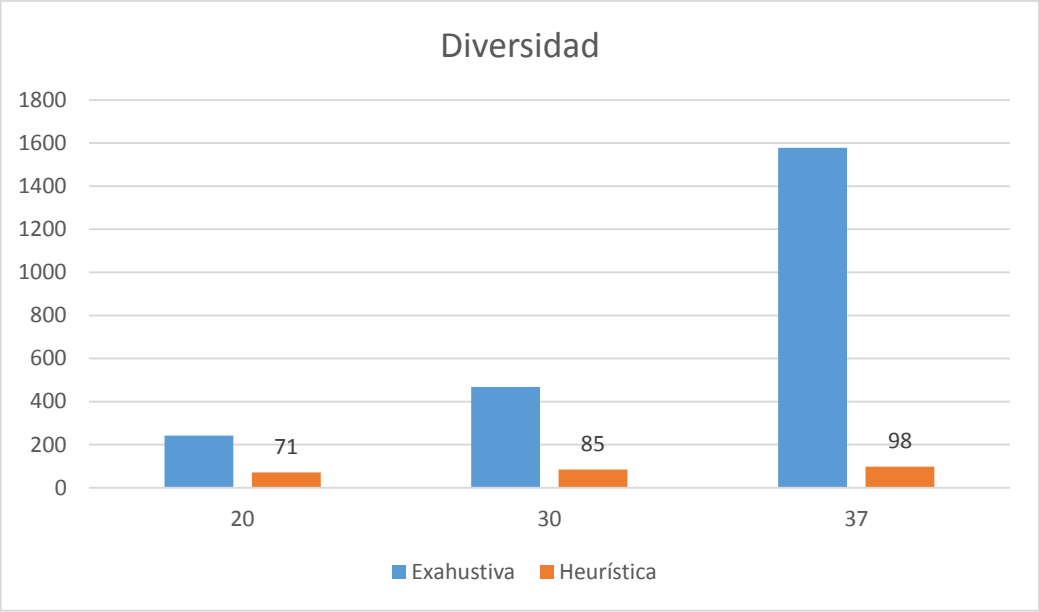

Figura 3.17 Tiempo de ejecución en el panel *Diversity Measures*

En el caso de la figura anterior se observa que a medida que aumenta la cantidad de clasificadores (20, 30, 37) la búsqueda exhaustica consume mucho mayor tiempo de ejecución (medido en segundos) que la búsqueda heurística; específicamente en el panel *Diversity Measures*, es decir, cuando se están calculando las combinaciones de clasificadores más diversos.

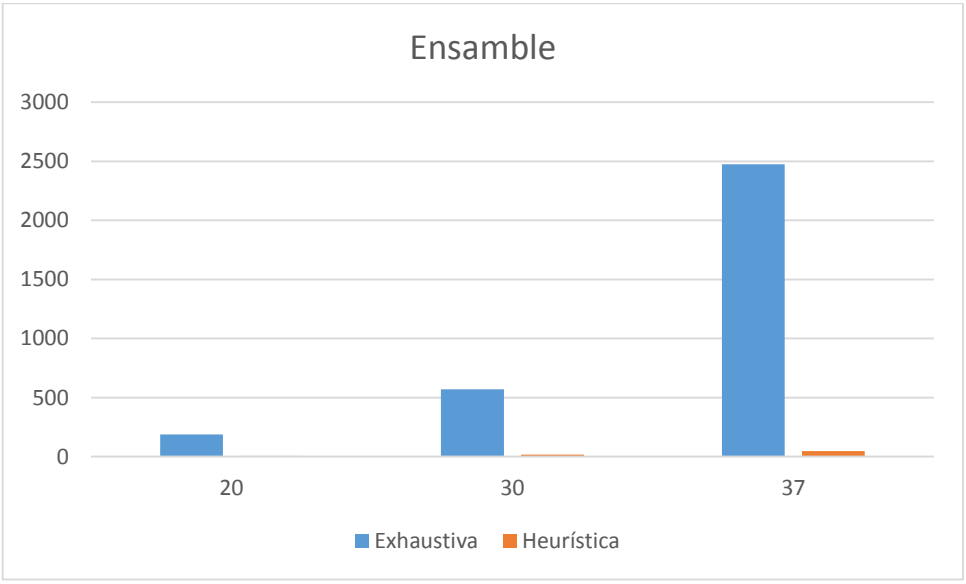

Figura 3.18 Tiempo de ejecución en el panel *Ensamble Methods*

En la figura anterior para el panel *Ensamble Methods* se puede observar la misma conclusión de la Figura 3.17 Tiempo de ejecución en el panel *Diversity Measures*. El tiempo de ejecución de la búsqueda exhaustiva es mucho mayor que la búsqueda

heurística; donde apenas se pueden apreciar las barras correspondientes a sus soluciones.

3.5 Modificaciones en la interfaz visual del software *DASDE v1.0.*

La herramienta "DASDE®" versión 2.0 como se ha mencionado en epígrafes anteriores está destinada a cualquier tipo de usuario, como estudiantes, especialistas o investigadores en química, biología, computación o ramas similares; permitiéndole al mismo distintas funcionalidades descritas en el epígrafe 2.1; la nueva versión del software incorpora la implementación de la meta heurística AG.

A continuación, se mostrarán las modificaciones realizadas en la interfaz visual del software para incorporar dicha meta heurística tanto para buscar las combinaciones de modelos diversos como para la obtención de las mejores combinaciones que superan a los modelos individuales.

3.5.1 Modificaciones en "*Diversity Measure*"

En la Figura 3.19 Vista de las búsquedas posibles se muestra la ventana que contiene los diferentes tipos de exploración (exhaustiva y heurística), implementándose en este trabajo la búsqueda heurística.

En el presente trabajo se añaden al panel las siguientes particularidades:

- ✓ Solo se puede seleccionar una opción de búsqueda (heurística o exhaustiva) a la vez.
- ✓ 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.
- ✓ En esta ventana se muestran activos los botones "***Stop***" y "***Run***".
- ✓ En la esquina inferior derecha se muestran 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***").

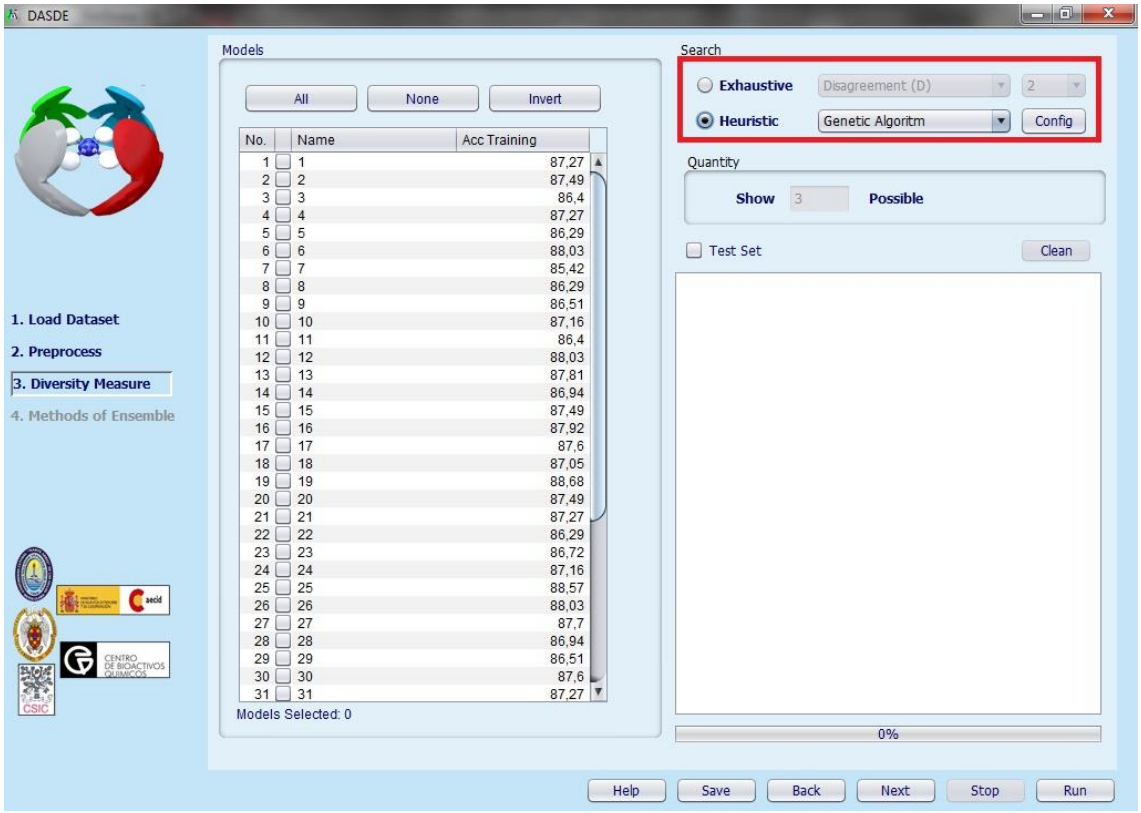

Figura 3.19 Vista de las búsquedas posibles

El usuario tiene la posibilidad de seleccionar cuál de las meta heurísticas desea aplicar: Algoritmos Genéticos (AG) o ACO. Al seleccionar *Genetic Algorithm* se debe presionar el botón "**Config**" y seguidamente se mostrará una ventana, la cual se muestra en la Figura 3.20 , donde se le brinda al usuario la posibilidad de introducir los valores de los parámetros para el funcionamiento de la meta heurística; permitiendo calcular la diversidad de acuerdo a distintas medidas.

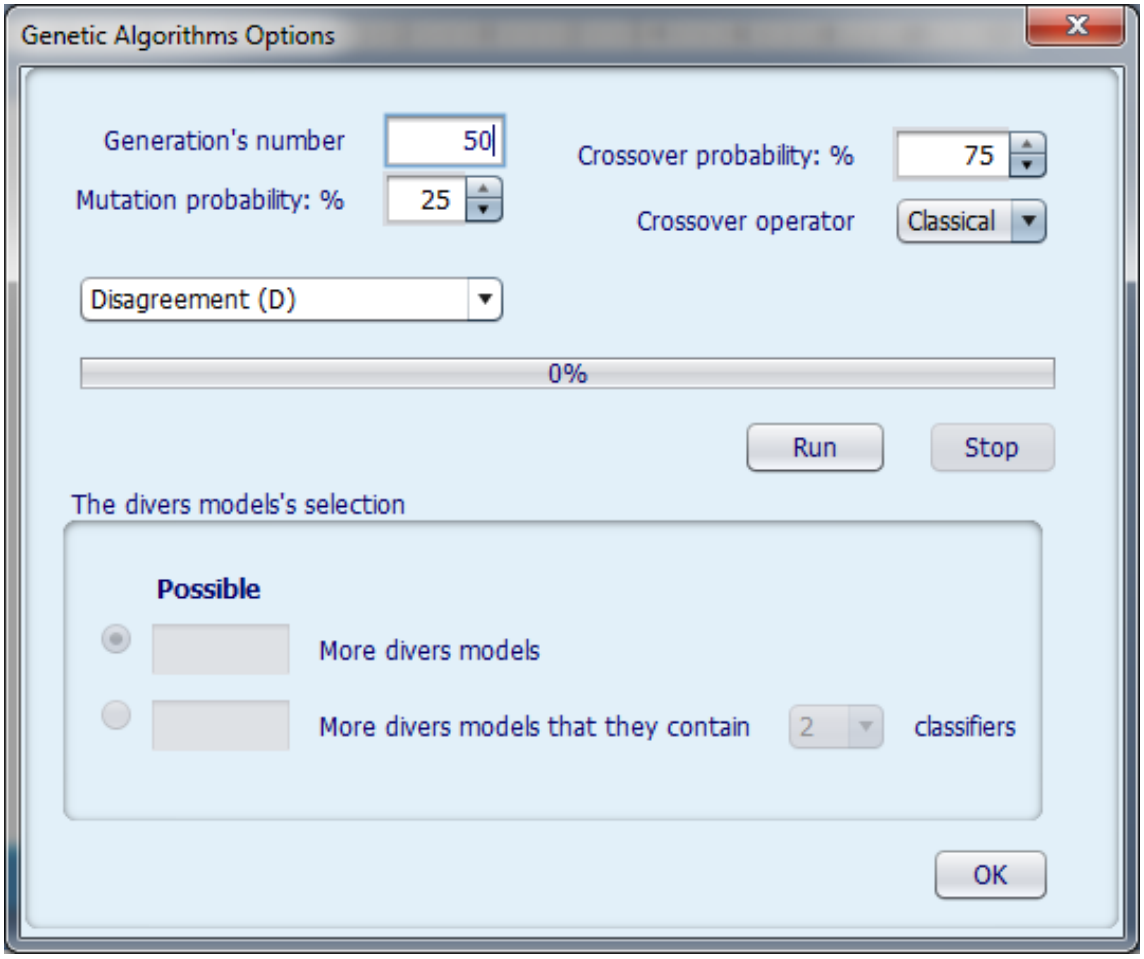

Figura 3.20 Ventana donde se insertan los parámetros de la cconfiguración del AG

Una vez introducidos los parámetros necesarios se presiona el botón "**Run**" para encontrar las posibles soluciones. Luego del proceso de búsqueda, se habilitan las opciones que permiten al usuario obtener la cantidad deseada de modelos diversos entre sí, sin importar el total de clasificadores incluidos en la solución, o teniendo en cuenta modelos que contengan un número específico de clasificadores introducido por el usuario, esto se muestra en el componente habilitado de la parte inferior de la Figura 3.21 .

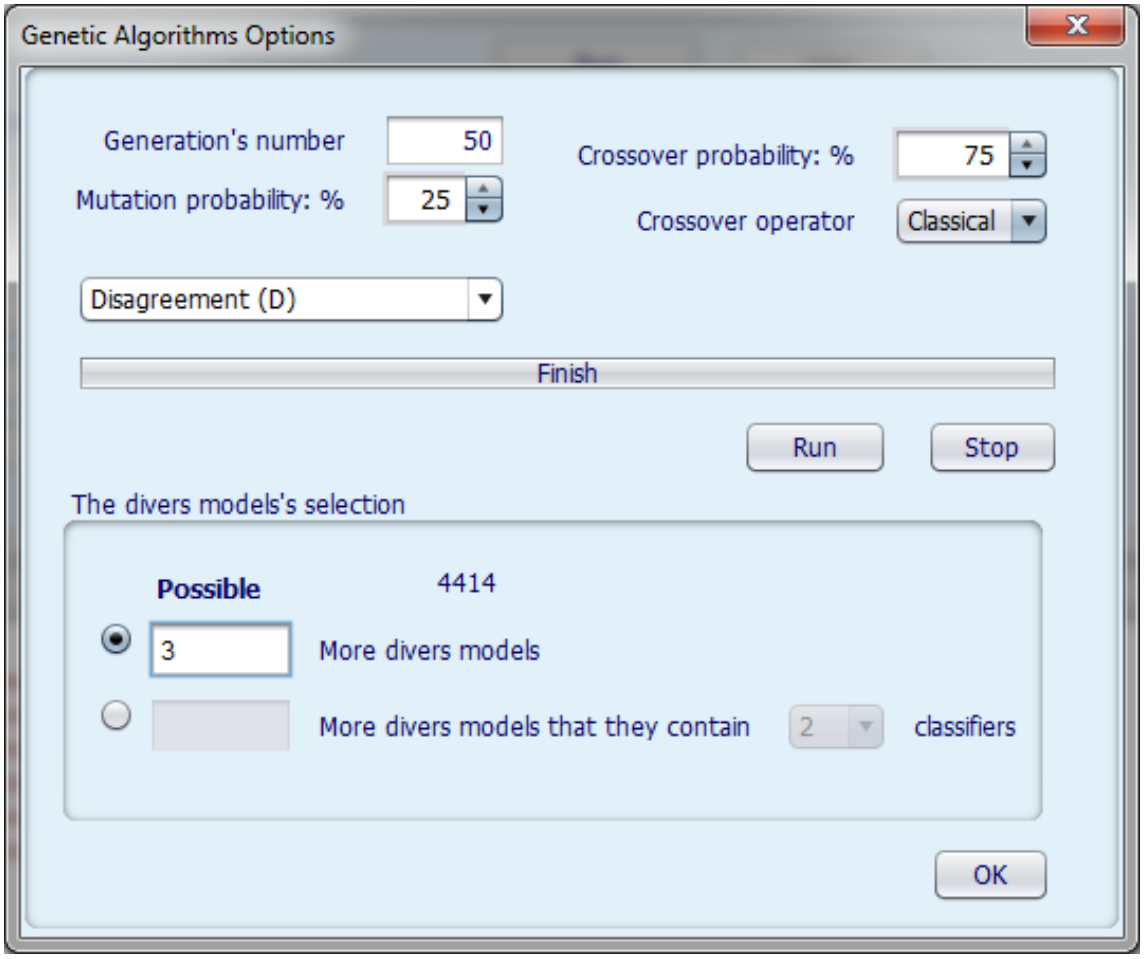

Figura 3.21 Ventana donde se insertan los parámetros de la configuración del AG con componente habilitado que permite definir las opciones de salida de las soluciones.

Posteriormente de haber llenado todas las opciones y al presionar en el botón "**Ok**" quedan listos los modelos más diversos para ser mostrados cuando se presione el botón "**Run**" del panel *Diversity Measure*, donde se puede apreciar cómo se muestran los resultados en la Figura 3.22 Salida de Diversity Measure.

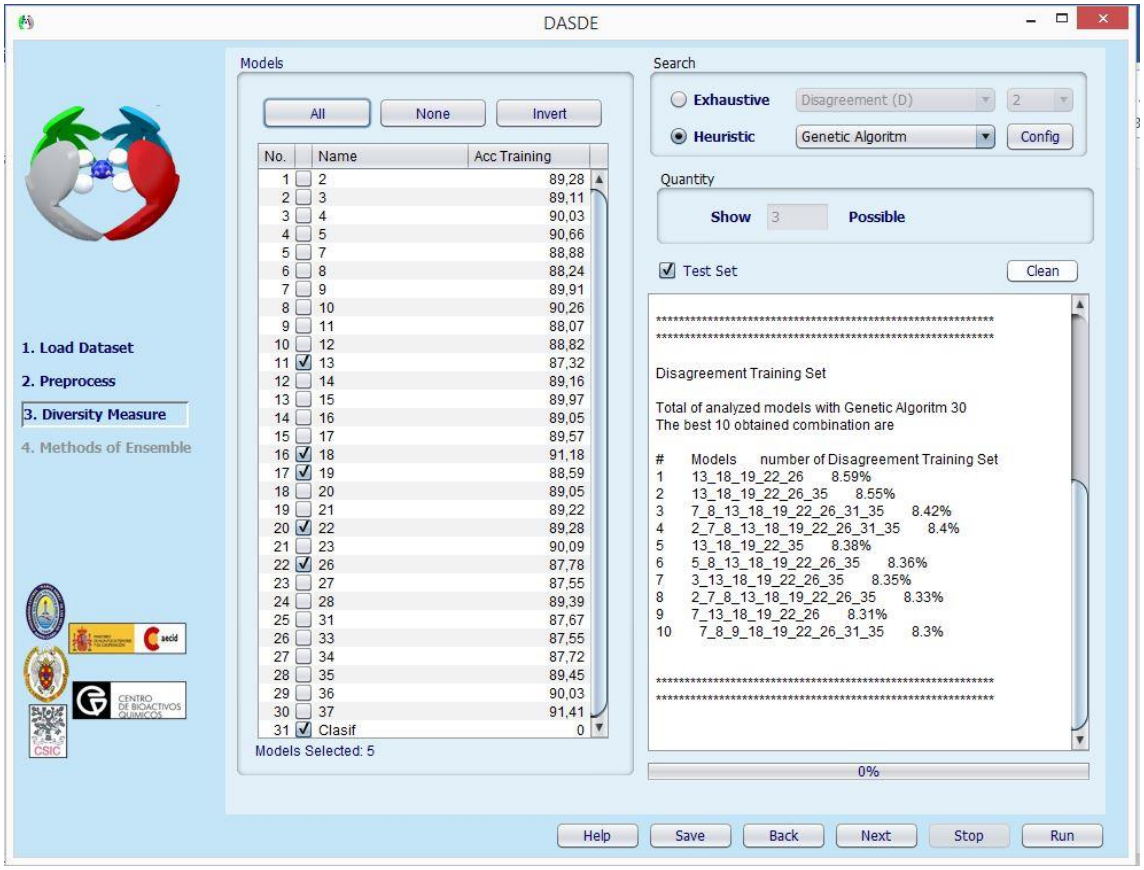

Figura 3.22 Salida de Diversity Measure

3.5.2 Panel *Ensemble Methods*

Para realizar el ensamble de los modelos individuales el usuario puede usar igualmente diferentes tipos de exploración (exhaustiva y heurística), implementándose en este trabajo la búsqueda heurística.

Similar que al paso anterior, al seleccionar *Genetic Algorithm* seguidamente se muestra una ventana (Figura 3.**23** Configuración y selección de la salida de los modelos ensamblados), en la cual se podrá introducir los valores de los parámetros necesarios como: cantidad de generaciones, el porcentaje de cruzamiento, mutación, así como, los operadores de cruzamiento tanto clásico como uniforme y las reglas de combinación para las salidas de los clasificadores (se pueden seleccionar varias). Una vez ejecutado el algoritmo mediante el botón "**Run**" se le brinda al usuario distintas opciones para seleccionar los mejores modelos ensamblados.

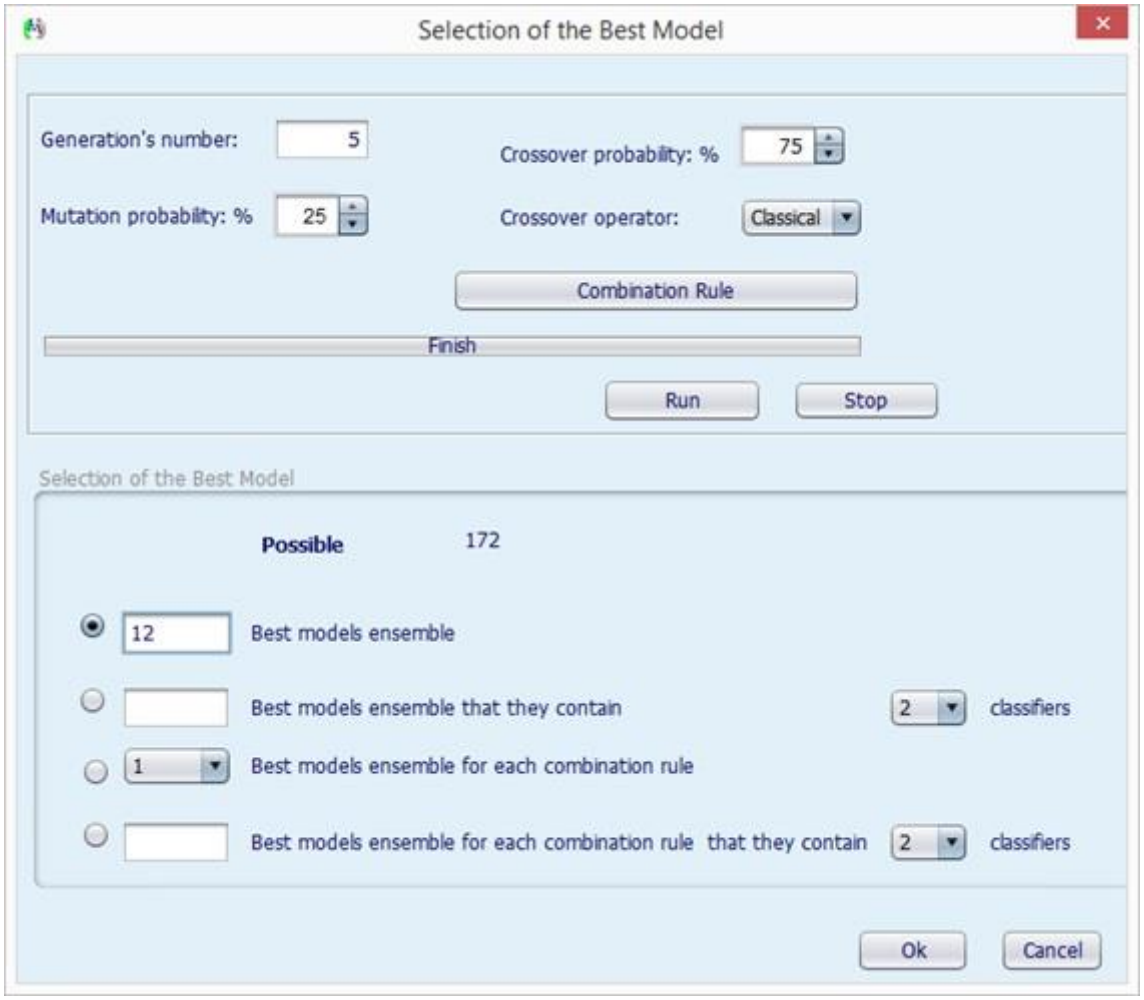

Figura 3.23 Configuración y selección de la salida de los modelos ensamblados

Al concluir en esta ventana seleccionando la forma de mostrar la salida de los resultados se presiona el botón "**Ok**" y las combinaciones obtenidas se muestran ordenadas según su exactitud luego de presionar el botón "**Run**" del panel *EnsembleMethods*.

Los nuevos modelos ensamblados se agregan a la lista de modelos individuales, mostrando además el valor de su exactitud. En la ventana de la parte inferior derecha de la Figura 3.**24** Vista de se muestra un resumen de los resultados obtenidos.

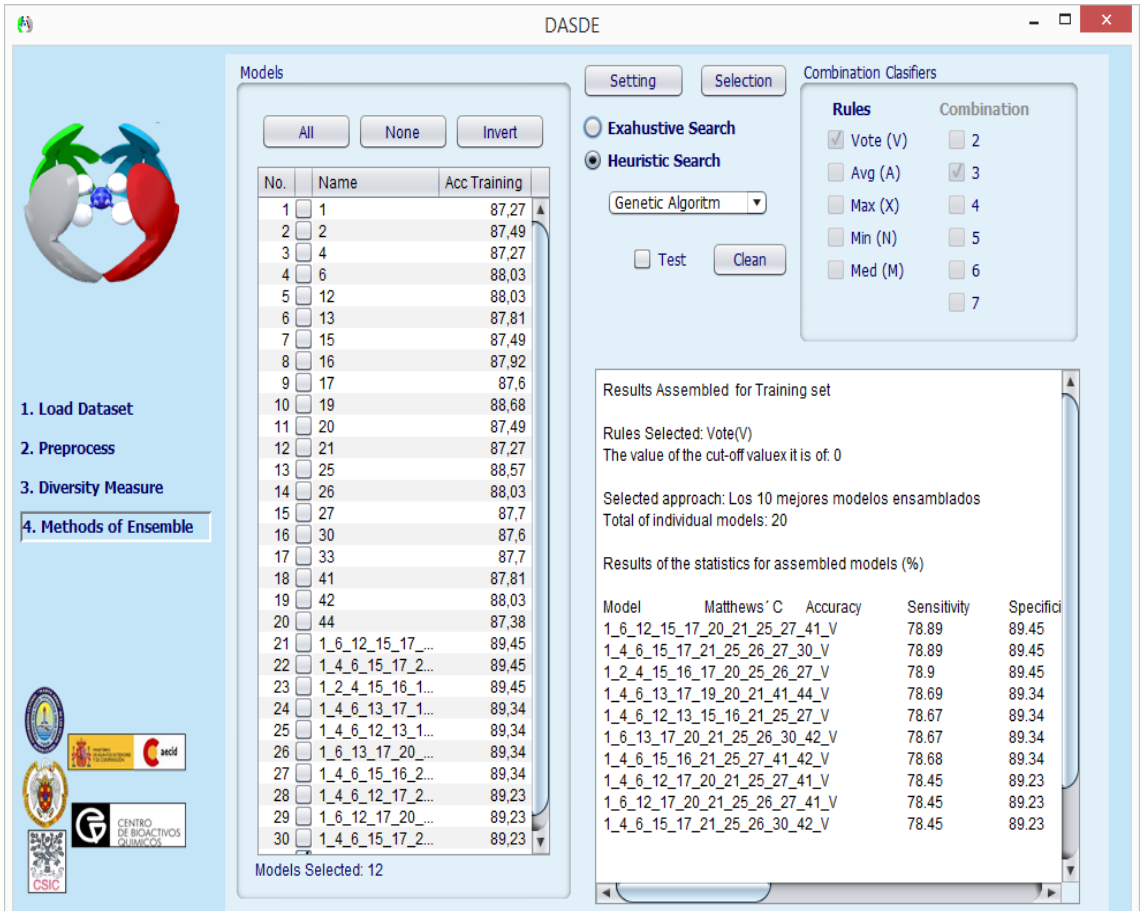

Figura 3.24 Vista de los resultados obtenidos

3.6 Consideraciones finales del capítulo

En el presente capítulo se muestran dos aplicaciones quimio-bioinformáticas con datos reales, en ambas se obtuvieron buenos resultados con el software en menor tiempo de ejecución, además se exponen las modificaciones principales en la interfaz visual del software DASDE v2.0 especificando las nuevas funcionalidades y forma de trabajar con el mismo.

CONCLUSIONES

1. Se implementaron dos medidas de diversidad no pareadas en el panel *"Diversity Measure"* del software DASDE v2.0: Desacuerdo entre expertos y Diversidad generalizada
2. Se incorporó al software DASDE v2.0 un módulo que permite el uso de la meta heurística Algoritmo Genético, para obtener combinaciones de clasificadores diversos.
3. Se incorporó al software DASDE v2.0 un módulo que permite el uso de la meta heurística Algoritmo Genético para obtener combinaciones de clasificadores que superen la mejor exactitud individual.
4. Para la validación de la propuesta se aplicó la meta heurística implementada en dos bases de datos reales que determinan la actividad de compuestos químicos relacionados con dos enfermedades: *malaria y antinflamatoria,* donde se obtuvieron combinaciones diversas de clasificadores que superaban la mejor exactitud individual, es decir, se obtienen soluciones que ayudan a resolver mejor la predicción de la actividad de los compuestos químicos.
5. Se demuestra que usando la búsqueda heurística se obtienen buenas soluciones en un tiempo mucho menor, lo cual agiliza considerablemente el proceso de descubrimiento y posterior desarrollo de un medicamento.

RECOMENDACIONES

1. Los resultados obtenidos se podrían utilizar en la corroboración experimental *in vivo* para la predicción de nuevos compuestos frente a las enfermedades de las bases de casos analizadas.
2. Validar el uso de la meta heurística implementada con otras bases de casos reales del campo de la quimio-bioinformática.

REFERENCIAS BIBLIOGRÁFICAS

Antonisse, J., 1989. A new interpretation of schema notation that overturns the binary encodeding constraint. In *Proceedings of the Third International Conference on Genetic Algorithms*. George Mason University, USA, pp. 86–91.

Bello, R. et al., 2001. Aplicaciones de la IA, Santa Clara.

Bonet Cruz, I., 2008. *Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial*. Santa Clara: Universidad Central Marta Abreu de las Villas.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, pp.24, 123–140.

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5–32.

Brown, G. & Kuncheva, L.I., 2010. " Good " and " Bad " Diversity in Majority Vote Ensembles. , pp.124–133.

Bulacio, P.., 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.

Cisneros Matos, C.A., 2007. *Selección/identificación asistida por computadora de nuevos compuestos líderes con actividad anti-inflamatoria*. Universidas Central "Marta Abreu" de las Villas.

Cunningham, P. & Carney, J., 2000. Diversity versus Quality in Classification Ensembles Based on Feature Selection. In *11th European Conference on Machine Learning*. Trinity College, Dublin, pp. 109–116.

Dietterich, T.G., 2000. "Ensemble methods in machine learning."

Díez-pastor, J.F. et al., 2015. Random Balance : Ensembles of variable priors classifiers for imbalanced data. , 85, pp.96–111.

Dorigo, M. & Birattari, M., 2010. Ant colony optimization. In *Encyclopedia of Machine Learning*. Springer, pp. 36–39.

Dowsland, K.A. & Adenso-Díaz, B., 2003. Diseño de Heurísticas y Fundamentos del Recocido Simulado. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7(19), pp.93–102.

Escalona, J.C., Carrasco, R. & Padrón, J.A., 2005. Introducción al diseño de Fármacos J.C. Escalona, R. Carrasco, J. A. Padrón.

Eshelman, L.J., Caruana, R.A. & Schaffer, J.D., 1989. Biases in the crossover

landscape. In *Proceedings of the third international conference on Genetic algorithms*. Morgan Kaufmann Publishers Inc., pp. 10–19.

Fernández, A.D. et al., 1996. Optimización heurística y redes neuronales. *ed. Paraninfo SA, Madrid, España*.

Fleiss, J.L., 1981. *Statistical Methods for Rates and Proportions* 2nd ed., New York.

Giacinto, G. & Roli, F., 2001. Design of effective neural network ensembles for image classification purposes. *Image vision and computing journal*, 19, pp.699–707.

Glover, F. & Melián-Batista, B., 2003. Búsqueda Tabú. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7(19), pp.29–48.

Goldberg, D.E., 1990. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5, pp.139–167.

Goldberg, D.E., 1989. Sizing populations for serial and parallel genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., pp. 70–79.

Hansen, L. & Salomon, P., 1990. "Neural Networks Ensembles." *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, pp.993–1001.

Herrera, F., Lozano, M. & Verdegay, J.L., 1998. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial intelligence review*, 12(4), pp.265–319.

Holland, J.H., 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*, Massachusetts, USA: U Michigan Press.

Kennedy, J., 2010. Particle swarm optimization. In *Encyclopedia of Machine Learning*. Springer, pp. 760–766.

Kohavi, R. & Wolpert, D.H., 1996. Bias Plus Variance Decomposition for Zero-One Loss Functions in Machine Learning. In *Machine Learning: Proceedings of the Thirteenth International Conference*. Los Altos, California, pp. 275–283.

Koza, J.R., 1992. Genetic programming.

Krawczyk, B. & Woźniak, M., 2014. Diversity measures for one-class classifier ensembles. *Neurocomputing*, 126, pp.36–44.

Kuncheva, L.I., 2013. A Bound on Kappa-Error Diagrams for Analysis of Classifier Ensembles. , 25(3), pp.494–501.

Kuncheva, L.I. & Jain, L.C., 2000. Designing classifier fusion systems by genetic

algorithms. *Evolutionary Computation, IEEE Transactions on*, 4(4), pp.327–336.

Kuncheva, L.I. & Rodríguez, J.J., 2014. A weighted voting framework for classifiers ensembles. , pp.259–275.

Kuncheva, L.I. & Whitaker, C.J., 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51, pp.181–207.

Kuncheva, L.I.., 2004. Diversity in Classifier Ensembles. In *Combining Pattern Classifiers: Methods and Algorithms*. New Jersey: John Wiley & Sons, Inc., pp. 295–327.

Machado Tugores, Y., 2013. *Tamizaje farmacológico en la búsqueda de potenciales fármacos antimaláricos integrando nuevos modelos in silico y corroboración experimental*. Universidad Complutense de Madrid.

Marrero-Ponce, Y. 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), pp.245–265. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=16475921.

Marrero-Ponce, Y. et al., 2008. Bond-based linear indices in QSAR: computational discovery of novel anti-trichomonal compounds. *J Comput Aided Mol Des*, 22(8), pp.523–540. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=18483767.

Marrero-Ponce, Y. 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), pp.9–26. Available at: <Go to ISI>://000262953100002.

Marrero-Ponce, Y. et al., 2006. Predicting antitrichomonal activity: a computational screening using atom-based bilinear indices and experimental proofs. *Bioorg Med Chem*, 14(19), pp.6502–6524. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=16875830.

Martí, R., 2002. Procedimientos Metaheurísticos en Optimización Combinatoria.

Meneses Gómez, M. & Cedré Gutiérrez, D., 2013. *Ambiente para la búsqueda de*

combinaciones de modelos bases en multiclasificadores de quimio-bioinformática. Universidad Central "Marta Abreu" de Las Villas.

Meneses-Marcel, A. 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), pp.3838–3843. Available at: <Go to ISI>://000231186000005.

Meneses-Marcel, A. et al., 2008. New antitrichomonal drug-like chemicals selected by bond (edge)-based TOMOCOMD-CARDD descriptors. *J Biomol Screen*, 13(8), pp.785–794. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=18753687.

Morales, A., 2014. Construcción de sistemas multiclasificadores usando Algoritmos Genéticos y medidas de diversidad.

Osman, I.H. & Kelly, J.P., 1996. *Meta-heuristics: theory and applications*, Springer.

Partalas, I., Tsoumakas, G. & Vlahavas, I., 2009. Pruning an ensemble of classifiers via reinforcement learning. *Neurocomputing*, 72(7), pp.1900–1909.

Partridge, D. & Krzanowski, W., 1997. Software diversity: practical statistics for its measurement and exploitation. *Information and Software Technology*, 39, pp.707–717.

Radcliffe, N.J., 1992. Non-Linear Genetic Representations. In *Parallel Problem Solving from Nature (PPSN)*. Amsterdam, pp. 261–270.

Rahman, A. & Tasnim, S., 2014. Ensemble Classifiers and Their Applications : A Review ABSTRACT : , 10(1), pp.31–35.

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

Santana, A., Soares, R.G.F. & Anne, M.P., 2006. A Dynamic Classifier Selection Method to Build Ensembles using Accuracy and Diversity.

Schapire, R.E., 1990. The strength of weak learnability. *Machine Learning*, 5(2), pp.197–227.

Segrera, F.S. & Moreno, G.M.N., 2006. *Multiclasificadores: Métodos y Arquitecturas.* España, Universidad de Salamanca: 47.

Skalak, D.B., 1996. The sources of increased accuracy for two proposed Boosting algorithms. In *Proc. American Association for Arti Intelligence, AAAI-96,*

Integrating Multiple Learned Models Workshop. pp. 120–125.

Syswerda, G., 1989. Uniform crossover in genetic algorithms. In *Proc. of the Third Int. Conf. on Genetic Algorithms*. Morgan Kaufmann Publishers, San Mateo.

Witten, I. & Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. *San Francisco, Diane Cerra.*

Wolpert, D., 1992. Stacked generalization. *Neural Networks*, pp. 5, 241–259.

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