UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD # 6

TÍTULO: Propuesta de Algoritmos para la Reducción de Instancias

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE

INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORES:

Kalianny Laffita Nicot

Yaima Mariño Zayas

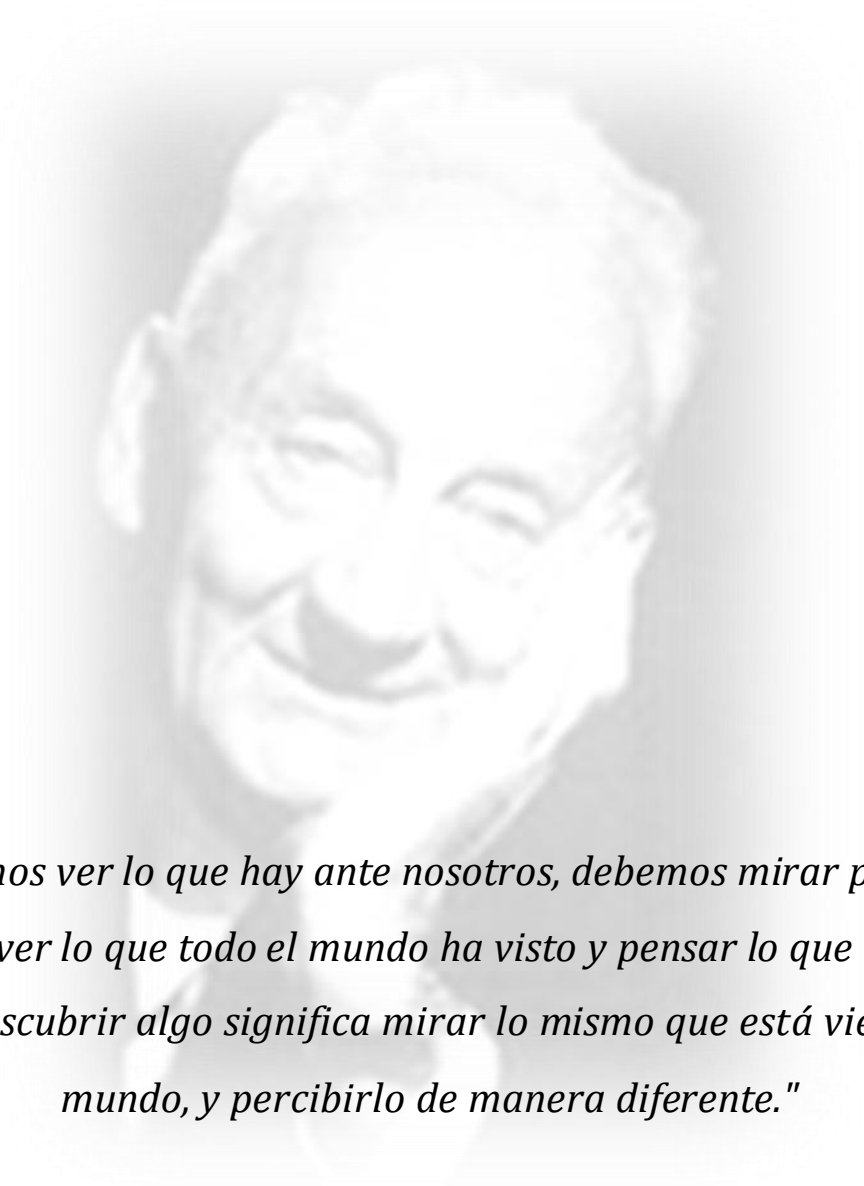
TUTORES:

Dr. Ramón Carrasco Velar

MSc. Yaikiel Hernández Díaz

Ciudad de La Habana, Cuba
Junio, 2010
"Año 52 de la Revolución"

PENSAMIENTO



"Si queremos ver lo que hay ante nosotros, debemos mirar para atrás. Investigar es ver lo que todo el mundo ha visto y pensar lo que nadie más ha pensado. Descubrir algo significa mirar lo mismo que está viendo todo el mundo, y percibirlo de manera diferente."

Albert Szent-Györgyi de Nagyrápol

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firman la presente a los ____ días del mes de _____ del año _____.

Kalianny Laffita Nicot

_____
Firma del Autor

Yaima Mariño Zayas

_____
Firma del Autor

Dr. Ramón Carrasco Velar

_____
Firma del Tutor

MSc. Yaikiel Hernández Díaz

_____
Firma del Tutor

DATOS DE CONTACTO

TUTORES:

Dr. Ramón Carrasco Velar

Doctor en Ciencias Químicas

Profesor Asistente

E-mail: rcarrasco@uci.cu

MSc. Yaikiel Hernández Díaz

Máster en Bioinformática

Profesor Instructor

E-mail: yhernandezd@uci.cu

AGRADECIMIENTOS

De Kalianny

Reconocer a todos los que nos han ayudado es un acto de generosidad. Uno puede devolver un préstamo de oro, pero está en deuda de por vida con aquellos que son amables. Quiero agradecer:

A mis padres y hermanos, que me han dado todo el cariño del mundo y han sabido guiarme, apoyarme, cuidarme y confiar siempre en mí.

A mi familia, por ayudarme a construir una vida llena de alegría y paz.

A mis tutores, en especial a Yaikiel Hernández, por ser exigente, responsable, por guiarnos y estar disponible siempre, incluso cuando no tenía tiempo.

A mis programadores favoritos, Luis, Dairon, Roberto y Edel, porque sin ustedes no lo hubiéramos logrado.

A mi novio Yanio Cueria Samón, por apoyarme y preocuparse en estos últimos meses tan difíciles, y sobre todo, por hacerme tan feliz.

A mis amigos baracoenses, que se han mantenido cerca de mí en los buenos y malos momentos.

A los amigos que me ha dado la UCI, en especial, las mejores amigas que pudiera tener, Yaima Mariño Zayas y Ludmila García Campos, por todos los momentos que vivimos juntas, por el apoyo inmenso que recibí de ellas, y más que todo, por ser hermanas para mí y permitirme serlo para ellas.

A mi primer grupo 6106, por ser mi apoyo en los primeros años, por darme buenos amigos y los mejores recuerdos.

A toda la gente linda que compartió la beca conmigo, en especial al apartamento 76203 y al actual 76101, por todas las cosas lindas compartidas y convertirla en una casa para mí.

A todos lo que de una forma u otra, han confiado en mí, me han permitido entrar en sus vidas y me han convertido en parte de ella.

He comprendido que solo en la agonía de despedirnos, somos capaces de comprender la profundidad de nuestro amor.

Gracias…

AGRADECIMIENTOS

De Yaima

Se me hace difícil decir tanto en tan pocas palabras, pero es tan grande el placer que se experimenta al encontrar personas agradecidas, que vale la pena arriesgarse a no ser un ingrato. Quiero agradecer:

A mi familia, que con su apoyo y confianza contribuyeron a que este sueño se hiciera realidad, en especial a mi hermano por su preocupación y a mis padres, podría hacer una tesis solo para agradecerles. Gracias por guiarme en la vida, todo el mérito de este trabajo se lo debo a ustedes.

A Kalianny, primero porque sin su ayuda no hubiese sido posible realizar este trabajo. La amistad es un alma que habita en dos cuerpos, un corazón que habita en dos almas. Gracias por ser mi amiga y creer en mí.

A mis tutores, en especial a Yaikiel, por atendernos hasta en sus momentos más complicados, por su entrega y dedicación. No puedo decir otra cosa que gracias y gracias.

A los profesores del tribunal, porque gracias a sus señalamientos pudimos presentar este trabajo con la calidad que tiene.

A Luis Grabiel, Dairon, Edel y Roberto, por caernos del cielo cuando pensábamos que todo estaba perdido y estar ahí para ayudarnos cada vez que lo necesitamos.

A los profesores que tuve durante los 5 años de la carrera, por estar presentes cada vez que tuve alguna duda y transmitirme de su sabiduría para llegar a ser la Ingeniera que soy hoy.

A Ludmila, eres la primera amiga que me dio la UCI, la que mejor me conoce, con la que más he compartido y a la que debería agradecer por soportar mis malcriadeces durante tanto tiempo, pero la amistad no se agradece, se corresponde y mi gratitud es tan absoluta que las palabras sobran.

A los grupos 6106 por el inicio, al 6301 por el cambio y al 6501 por el presente. A todos, porque en ellos tuve la oportunidad de encontrar amigos como Yilianis, Surama, Elisa, Marisel, Themis, Geovanis, Salvador, Eneysi, Yuly y muchos más, que sé que a pesar de las distancias, perdurarán.

A otros amigos que me dio la UCI y con los que compartí momentos de tristezas y alegrías. A Papín, Ale, Leo, Silvio, Osniel, Rayko y con quien de momento la alegría se convirtió en catástrofe, "Quino, jajaja…tú sabes".

A todas las niñas del apartamento 76101, Leysi, Roxy, Yubi, Yuya y al piquete de las "Lunas"; Yadi, Ludmi, Kaly, Tati y Ana, a todas por la confianza y la amistad que adquirimos en la convivencia.

En fin, a todos aquellos que de una forma u otra ayudaron a hacer realidad mi sueño, sepan que nunca digo adiós a nadie, nunca dejo que las personas más cercanas a mí se marchen. Me las llevo conmigo a donde vaya.

Gracias…

DEDICATORIA

De Kalianny

Por enseñarme a ser paciente, a confiar, a apreciar y demostrarme que no importa que sea lo que tienes, sino a quien tienes en la vida, dedico este trabajo:

A mis padres, pues son mi mayor orgullo. Por estar siempre a mi lado y convertirme en la mujer que soy hoy.

A mis hermanos, los mejores del mundo, por apoyarme y cuidarme siempre.

A mis abuelos, con los que no puedo compartir mi vida y nunca se alejarán de mi corazón.

A mis abuelitas, por llenar de bellos momentos cada espacio de mí.

A mis tíos, en especial Benilda, por ser una madre para mí, y mi tío Félix, por aconsejarme, apoyarme y por sobre todas las cosas, por ser amigo.

A mis primos, por quererme tanto y tenerme tan presente en sus vidas.

A mi familia, por confiar en mí y acompañarme en el camino que escogí.

A mis amigos, por tenderme siempre la mano cada vez que lo necesitaba y motivarme a seguir, por ser una parte importante de los buenos y malos momentos que me ha tocado vivir.

A la UCI, por formarme como profesional. Por ser una inmensa casa llena de nuevos amigos y buenos recuerdos.

A la Revolución, por ser la creadora de la UCI y darme la oportunidad de estudiar en ella.

Los quiero…

DEDICATORIA

De Yaima

Lleva mucho tiempo llegar a ser la persona que quieres ser en la vida. Lo que soy hoy, se lo debo a mi familia. Por enseñarme a soñar, a vivir, a tener fe y a hallar amor en el mundo. Porque perdurará siempre la huella del camino enseñado, le dedico este trabajo:

A mi mamá, mi papá y mi hermanito, por ser el principal argumento de mi vida. No duden nunca que los quiero y que mi amor por ustedes es infinito y duradero como las estrellas en el cielo.

A la memoria de mis abuelos Esperanza y Conrado, y a los que están presentes, Felicia y Juan Ignacio, a los cuatro porque siempre los amaré y por ser los mejores abuelos del universo entero.

A mis tíos Jorge, Ignacio, Daniel y a mis tías Dinorah, Mayda, Milagros, Arelis y Nilda, por su preocupación e insaciable apoyo para que todo me saliera bien.

A mis primas Mayelín, por ser mi cámara de los secretos y Aymée, por su incondicionalidad. A mi primo Raudelis, por sus sabios consejos a pesar de su juventud y a los de más pura inocencia, Maylín, Ahmed, Carlos, Javier, Yanecita, Claire. A todos, porque de una forma u otra han estado ahí siempre para ayudarme a superar los momentos difíciles.

A mi Comandante en Jefe Fidel Castro Ruz una dedicatoria especial, pues gracias a su abnegado esfuerzo, podemos contar hoy con nuestra Revolución y con una universidad tan maravillosa como la UCI.

Los quiero…

RESUMEN

El presente trabajo muestra, la descripción e implementación de algoritmos evolutivos para la reducción de instancias en las muestras de la Plataforma para la Predicción de Actividad Biológica en Compuestos Orgánicos (alasGRATO). Actualmente las muestras calculadas en la plataforma, por la particularidad de los datos tratados, han tendido a disminuir la cantidad de variable e incrementar la cantidad de instancias haciendo que los valores de clasificación no sean eficientes. Como propuesta de solución a este problema se implementaron dos algoritmos genéticos, basados en la selección de prototipos aplicada a la selección de conjuntos de entrenamiento, con el fin de obtener modelos predictivos de menor tamaño y mayor interpretabilidad. Estos son Aprendizaje Probabilístico basado en Poblaciones (*Population-Based Incremental Learning*) por sus siglas PBIL y Recombinación heterogénea de la selección elitista generacional cruzada y mutación cataclísmica (*Cross generational elitist selection Heterogeneous recombination and Cataclysmic mutation*) por sus siglas CHC. En los casos experimentados a partir de muestras reales se reducen las instancias aproximadamente en un 94% para PBIL y un 96% para CHC, no siendo significativa la diferencia entre los resultados alcanzados por ambos algoritmos. CHC se destaca por presentar el menor costo computacional. Se presenta además la vista lógica de la aplicación desarrollada, así como el uso de los patrones de diseño utilizados, como contribución a su posterior inclusión en la plataforma.

PALABRAS CLAVE

Algoritmos Evolutivos, Algoritmos Genéticos, CHC, PBIL, Reducción de Instancias, Selección de Prototipos

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLA

INTRODUCCIÓN

El uso de medicamentos para sosegar el dolor o sentirse físicamente mejor, está presente en el hombre desde sus orígenes, pues la necesidad de hallar solución a sus males ha sido siempre tan importante como el instinto de alimentarse o de sobrevivir (1). La farmacología[1]: *ciencia biológica que estudia las acciones y propiedades de las drogas o fármacos en los organismos vivos*, es precisamente una de las múltiples disciplinas científicas, que sin duda, es y será un referente para la investigación futura en la creación de los fármacos, así como el estudio de los mismos.

Los programas de salud creados en los países, tanto desarrollados como subdesarrollados, para procesar medicamentos y tratar enfermedades curables tales como: inmunodeficiencia, esquizofrenia, esclerosis múltiple, sinusitis, glaucoma, otitis, hipertensión, herpes genital, parálisis cerebral, entre otras, resultan muy complejos. En nuestros días, la medicina moderna ha alcanzado avances en la síntesis y creación de los medicamentos, aunque quedan enfermedades para las cuales no existe cura, sin embargo, se han desarrollado fármacos que alivian las consecuencias de las mismas y hacen posible una mejor calidad de vida de estas personas. Entre estas enfermedades encontramos la polio, conocida también como poliomielitis o parálisis infantil, la diabetes, el Síndrome de Inmunodeficiencia Adquirida (SIDA) producida por el Virus de la Inmunodeficiencia Humana (VIH), el asma y el cáncer, por solo citar algunos.

Los procesos de desarrollo de fármacos abarcan áreas de interés para numerosas disciplinas afines, como el fisiológico, químico, bioquímico, farmacéutico y la informática (2, pág. 1), que como herramienta de ayuda a la medicina es una realidad en auge. La informática ha introducido una nueva dimensión en el pensamiento humano, haciendo reales proyectos que parecían imposibles. Los avances en esta tecnología y en el desarrollo de redes de información han permitido fabricar fármacos en menor tiempo, de una forma ecológicamente compatible y a costos más reducidos (3). A pesar de ello, los costos adquisitivos son elevados y se hace muy difícil que países del tercer mundo, así como muchos países desarrollados, puedan acceder a los mismos.

De forma cada vez más profusa se van utilizando las ventajas de la informática en un entorno caracterizado por el aumento del número y la complejidad de los datos. En el mundo se han diseñado grandes bases de datos para el almacenamiento de la información que se obtiene en las investigaciones

[1] **Dra. Mayra Levy Rodríguez.** Farmacología. Su historia y desarrollo. Colectivo de autores. *Farmacología General.* s.l.: Ciencias Médicas, 2004.

de desarrollo de nuevos compuestos químicos para enfermedades que afectan a millones de personas de países en desarrollo. Debido al gran cúmulo de información que presentan dichas bases de datos, muchos investigadores se han dado a la tarea de diseñar algoritmos para hacer más cómodo y eficiente el trabajo de minería de datos. De ahí que surgen algoritmos con la finalidad de reducir el espacio muestral de estas cantidades de datos a costos y niveles de rendimiento razonables.

Debido al férreo bloqueo económico que se ha impuesto a nuestro país durante cinco décadas, el sistema de salud cubano se ha visto imposibilitado de adquirir medicamentos para tratar múltiples enfermedades. Ha tenido la necesidad de desarrollar centros científicos e investigativos, así como priorizar la industria cubana de producción de fármacos, con el objetivo de curar o tratar las enfermedades que presentan todos los pacientes, sin importar los recursos que tenga que invertir en ello. Pese a ser un producto no tradicional en las exportaciones cubanas, los fármacos constituyen para Cuba un pilar fundamental en su economía. El grueso de las exportaciones cubanas de medicamentos genéricos se destina a países del área, fundamentalmente Venezuela, Brasil, Nicaragua, Bolivia, Argentina y Colombia por valor de 188 millones de dólares. Por otra parte, los productos biotecnológicos se dirigen en su mayoría hacia países de Latinoamérica y otras áreas geográficas, estimándose que las exportaciones de estos medicamentos estén en el entorno de los 30 millones de dólares. Dichas exportaciones recogen tanto las ventas de Farmacuba que superan 140 millones de dólares, así como de las empresas del Polo Científico, es decir, Cimab, Vacunas Finlay, la Heber Biotec con 412 marcas farmacéuticas registradas en el mercado de productos biotecnológicos, así como de Laboratorios Dalmer (4, pág. 17).

La Universidad de las Ciencias Informáticas (UCI), como proyecto de la Batalla de Ideas y rectora de la informatización en el país, se encuentra inmersa en la producción de *software* para las diferentes esferas. Una de estas es la Bioinformática, perfil que despliega la Facultad 6 y en la que se desarrolla la plataforma alasGRATO. Dicha plataforma cuenta con una base de datos de gran tamaño, conformada por moléculas y sus descriptores asociados, los cuales son utilizados por los métodos de inteligencia artificial implementados en la misma para la predicción de actividad biológica, asociando esta a la estructura química.

Anteriormente, al procesar una cifra engrandecida de datos, existía un elevado consumo de los recursos de cómputo. Para darle solución a este inconveniente se hizo necesario la implementación de una herramienta que contribuyera a la reducción del espacio muestral de descriptores. El objetivo de esta aplicación era eliminar gran parte de la redundancia de información en la base de datos, mejorar la

eficiencia y costo computacional del establecimiento de los modelos y la realización de las predicciones (5, pág. 2-3). Surge entonces el módulo “Selección de variables”, en el que se logró reducir las variables, cumpliendo satisfactoriamente el objetivo trazado. Sin embargo, los valores de la clasificación no son eficientes debido a que las muestras actualmente calculadas, por la particularidad de los datos tratados, han tendido a disminuir la cantidad de variable e incrementar la cantidad de instancias, haciéndose necesario realizar una adecuada selección de las mismas.

En consecuencia, el **problema científico** se expone de la siguiente manera: ¿Cómo reducir las instancias en las muestras de la plataforma alasGRATO?

Se define como **objeto de estudio**: La selección de instancias.

El objeto de estudio delimita el **campo de acción**: Los algoritmos de búsqueda y evaluación para la selección de instancias.

Con el fin de solucionar el problema planteado se define como **objetivo general**: Proponer algoritmos de búsqueda y evaluación para la reducción de instancias en las muestras de la plataforma alasGRATO.

Para cumplir el objetivo general se establecen los siguientes **objetivos específicos**:

- Identificar algoritmos y metodologías aplicables a la reducción de instancias.
- Implementar los algoritmos seleccionados.
- Validar los algoritmos implementados a partir del clasificador Máquinas de Soporte Vectorial (MSV).

Con el afán de solucionar la situación problemática y dar cumplimiento a los objetivos trazados se establecieron las siguientes **tareas de la investigación**:

- Revisión bibliográfica acerca de los algoritmos de selección de instancias y de criterios de evaluación.
- Estudio de la metodología y herramientas para el desarrollo del sistema.
- Implementación de los algoritmos seleccionados para dar respuesta a la reducción de instancias.

- Aplicación de los diferentes algoritmos implementados para reducir las instancias en muestras de Cefalosporinas y Factor Esteroidogénico-1.
- Aplicación del clasificador MSV a muestras completas de Cefalosporinas y Factor Esteroidogénico-1.
- Aplicación del clasificador MSV a muestras con instancias reducidas de Cefalosporinas y Factor Esteroidogénico-1.
- Realización de pruebas no paramétricas para la comparación de los resultados.
- Construcción de un *plug-in* con los algoritmos seleccionados para la versión 1 de la plataforma alasGRATO.

Como **posibles resultados** se espera: *Plug-in* que contenga algoritmos que reduzcan las instancias aplicando técnicas de inteligencia artificial.

El presente documento está estructurado en Resumen, Introducción y tres capítulos de los cuales a continuación se sintetiza su contenido.

Capítulo # 1: Fundamentación Teórica

Contempla una descripción de los aspectos más importantes afines a la investigación, enunciando conceptos y definiciones que ayudan a la comprensión del tema. Se refleja además, el estado del arte de los métodos y técnicas utilizados en el mundo, así como los sistemas automatizados vinculados a estos métodos.

Capítulo # 2: Materiales y métodos

Se presentan en la sección de Métodos, una breve descripción de las características de las muestras. Los AE para la reducción del espacio muestral de forma vertical y se explican las peculiaridades de los mismos. Se definen las herramientas y el entorno de desarrollo que se utilizaron para la implementación y evaluación de los algoritmos seleccionados.

Capítulo # 3: Resultados y discusión

En este capítulo se muestra un modelo conceptual para explicar brevemente en qué consiste la aplicación implementada. Se presenta la vista lógica y los patrones de diseño empleados en la programación orientada a objetos. Se describen detalladamente los algoritmos implementados y la validación de los mismos a través de pruebas con datos reales, así como pruebas no paramétricas para comprobar el correcto funcionamiento de la aplicación.

Finalmente, se unen a lo anteriormente expuesto las Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y el Glosario de Términos.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se muestra una descripción de los aspectos más importantes de la investigación, donde se enuncian los conceptos y definiciones que hacen posible una mejor comprensión del tema. Se refleja además el estado del arte sobre las técnicas y métodos utilizados en el mundo, así como los sistemas automatizados vinculados.

1.1 Introducción a la Selección de Instancias (SI)

La reducción del conjunto inicial de datos mediante SI está basada en escoger tan solo las muestras más representativas de entre todo el conjunto, de tal forma que el conjunto seleccionado no pierda información. Reduciendo el conjunto inicial de datos se mejora tanto la complejidad en tiempo computacional, como los recursos de almacenamiento. La eliminación de instancias no produce una degradación de los resultados, debido a la no existencia en la información de muestras repetidas o ruido. Cada instancia presenta un cierto grado de libertad suficiente, pues al reducir el número, en algunos casos elimina el sobre entrenamiento de las técnicas de aprendizaje supervisado (7, pág. 28). Los algoritmos de SI se ven afectados por el tamaño del conjunto de datos sobre el cual se aplican. Dado que las técnicas de SI presentan órdenes de eficiencia superiores a $O(n^2)$, siendo n el número de muestras del conjunto, los requerimientos tanto de tiempo de cálculo como de recursos necesarios aumentan considerablemente con el tamaño del conjunto de entrada (6, pág. 4). La SI se puede llevar a cabo siguiendo diferentes vías o estrategias como se muestra en la Figura 1.

Figura 1.1: Estrategias de Selección de Instancias

A continuación se describe cada una de las estrategias de SI mostradas en la figura (6, pág. 28-31):

- **Muestreo**

Consiste en escoger un subconjunto de instancias del conjunto original, mediante un proceso aleatorio de selección caracterizado porque cada muestra presenta una probabilidad de ser escogida. Los diferentes modelos de muestreo existentes son:

  - Muestreo Aleatorio
  - Muestreo Estratificado
  - Muestreo por Agrupamiento
  - Muestreo Sistemático
  - Muestreo Doble
  - Muestreo Enlazado
  - Muestreo Inverso
  - Muestreo Progresivo

Cada tipo de muestreo tiene características propias, con variantes que hacen posible que cualquier elemento del conjunto tenga la misma probabilidad de ser seleccionado e incluso que una instancia pueda ser seleccionada múltiples veces. En algunos casos, el conjunto final seleccionado estará formado por la unión de las muestras en cada una de las capas y otros repiten continuamente el proceso de selección del subconjunto solución, hasta que este satisface una serie de condiciones específicas.

- **Combinación de Clasificadores**

Las técnicas desarrolladas de clasificación por votación o conjunta, se dividen en dos grupos: los que cambian la distribución de los ejemplos de entrenamiento (*Boosting*) y los que no (*Bagging*). Los *Boosting* emplean todas las instancias en cada evaluación, manteniendo un peso para cada instancia del conjunto de entrenamiento que refleja su importancia. Por otra parte, los *Bagging* generan conjuntos de entrenamiento mediante la selección con reemplazamiento sobre el conjunto inicial y combina los conjuntos obtenidos mediante votación para construir una decisión común. La idea básica es correr un algoritmo de inducción varias veces, combinarlos y obtener un mejor resultado. De esta forma, se pretende combinar las decisiones tomadas por diferentes clasificadores para confeccionar una decisión conjunta a partir de ellas, con la que se efectúa la selección.

- **Aprendizaje Activo**

El proceso de Aprendizaje Activo (*Active Learning*) presenta como diferencia destacable frente a las anteriores técnicas citadas, el hecho de que el subconjunto final seleccionado es dinámico. Conforme se van clasificando nuevas instancias, serán agregadas al conjunto de entrenamiento las que aporten mayor información, mejorando así, las prestaciones que este proporciona.

- **Selección de Prototipos (SP)**

Su objetivo es seleccionar un conjunto de muestras que mejore la capacidad de predicción de un clasificador, basado en la regla del vecino más cercano, con el fin de eliminar aquellas muestras ruidosas o redundantes y acelerar el proceso de clasificación.

Se elige esta estrategia, pues mediante la misma se puede llevar a cabo la selección de los conjuntos de entrenamiento, prometedora para obtener modelos predictivos con alta precisión e interpretabilidad. Esta técnica, desarrollada dentro de la selección evolutiva para la reducción de datos, permite abordar el problema de escalado que puede aparecer cuando se manejan conjuntos de entrenamiento de gran tamaño y por tanto, una solución factible al problema que presenta la plataforma (6, pág. 31).

1.2 Selección Evolutiva de Instancias para la Reducción de Datos

1.2.1 Estrategias de Selección de Instancias: Clasificación basada en Prototipos y Selección de Conjuntos de Entrenamiento

La SP se puede llevar a cabo siguiendo dos objetivos diferentes, considerando mejorar la capacidad de predicción del clasificador basado en la regla del vecino más cercano (Clasificación), o bien la selección de conjuntos de entrenamiento a partir de los que se pueden obtener modelos descriptivos o predictivos posteriormente (6, pág. 41-42). A continuación se describen dichas estrategias.

- **Clasificación basada en la Selección de Prototipos mediante la técnica del vecino más cercano**

En este caso se desea mejorar el comportamiento del clasificador basado en el vecino más cercano, mediante la selección de las muestras de entrenamiento con mayor representatividad y capacidad de generalización. Figura 2.

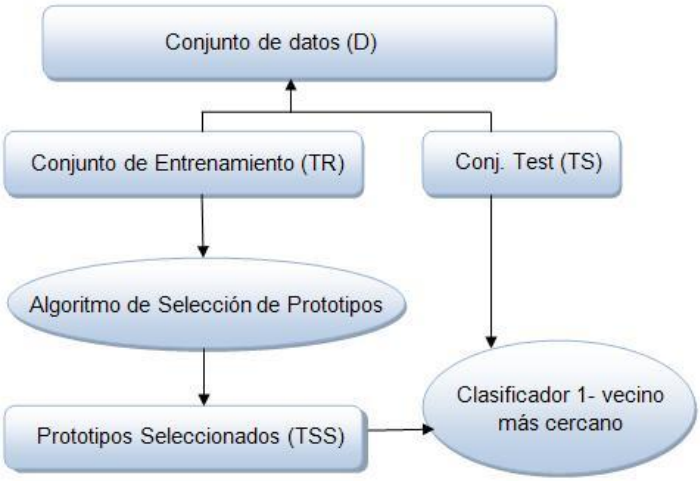


Figura 1.2: Selección de Prototipos aplicada a clasificación

El conjunto inicial (D) se divide en dos, uno sobre el que se llevará a cabo la selección (TR) y otro que será empleado para validarla (TS). Sobre el conjunto TR se aplica el algoritmo de selección para obtener el conjunto de prototipos que se emplearán como conjunto de entrenamiento en el clasificador 1-vecino más cercano, empleando como conjunto de test a TS (6, pág. 42-43).

- **Selección de Conjuntos de Entrenamiento**

El objetivo perseguido es obtener aquel conjunto de instancias, al que denominamos Prototipos Seleccionados (TSS), que independientemente del algoritmo de aprendizaje que se le aplique para obtener el modelo, mantenga su representatividad y capacidad de generalización. Figura 3.

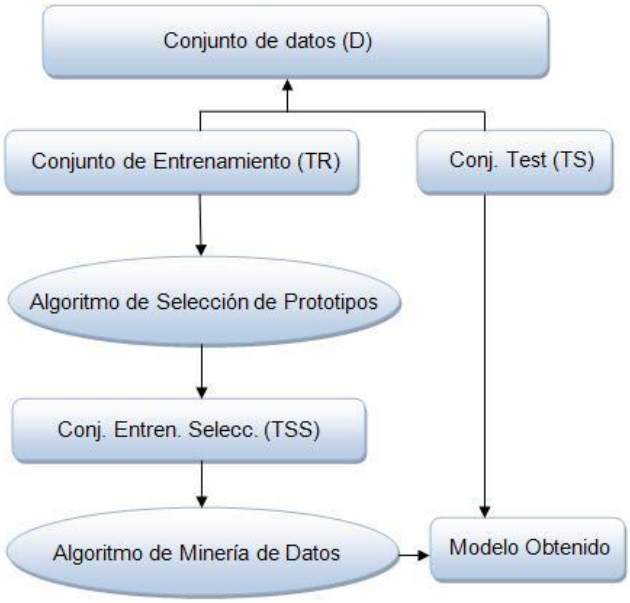


Figura 1.3: Selección de Prototipos aplicada a selección de conjuntos de entrenamiento

Del mismo modo que en el caso anterior, el conjunto inicial (D) se divide en dos, TR y TS. Sobre TR se aplica el algoritmo de selección para obtener TSS como conjunto de entrenamiento seleccionado. Este conjunto TSS se emplea como entrada para el algoritmo de minería de datos, y generará un modelo a partir del conjunto TSS de entrada que será validado empleando TS (6, pág. 43-44).

Ambas estrategias posibilitan que independientemente de los algoritmos que se le apliquen a los prototipos seleccionados conserven su información. La clasificación basada en la SP mediante la técnica del vecino más cercano, es un método independiente del dominio, no tiene restricciones del lenguaje y posee una alta capacidad de generalización y clasificación (7, pág. V). A pesar de esas características, se escoge la selección de conjuntos de entrenamiento pues se considera que presenta un mejor comportamiento ya que consigue los modelos predictivos de menor tamaño, compuestos por reglas con capacidades elevadas de predicción e interpretabilidad. Al mismo tiempo, el componente evolutivo optimiza la exploración en él, obteniendo porcentajes de acierto elevados (8, pág. 10-11).

1.2.2 Técnicas No Evolutivas de Selección de Instancias

Los métodos de SP son técnicas de SI que pretenden encontrar conjuntos de entrenamiento tales que ofrezcan los mayores porcentajes de clasificación empleando la regla del vecino más cercano (1-NN) (6, pág. 44).

A continuación se describirá cada una de las estrategias que se pueden llevar a cabo en el proceso de SP como se muestra en la figura 4.

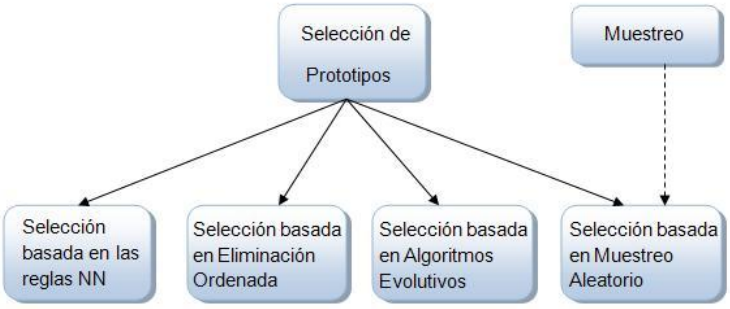


Figura 1.4: Estrategias de Selección de Prototipos

- **Selección basada en la regla del vecino más cercano (reglas NN)**

En este tipo de estrategia se encuentran todos aquellos métodos que fundamentan sus estrategias de selección en la regla del vecino más cercano. Esta regla se basa en clasificar una nueva instancia perteneciente a la clase correspondiente a su vecino (6, pág. 45-46).

- **Selección basada en eliminación ordenada**

En este se encuentran los algoritmos de la familia *DROP* (*Decremental Reduction Optimization Procedure*), los cuales están basados en la eliminación ordenada de instancias (6, pág. 46-47).

- **Selección basada en Muestreo Aleatorio**

Incluye métodos de muestreo que se adecúan a la SP (6, pág. 47).

- **Selección basada en Algoritmos Evolutivos (AE)**

En este caso la selección se lleva a cabo empleando mecanismos basados en la evolución natural que pueden ser utilizados para problemas de búsqueda y optimización (6, pág. 47).

Teniendo en cuenta que este tipo de selección maximiza la tasa de clasificación y minimiza el número de instancias obtenidas, además de ofrecer resultados interesantes cuando se emplean en la SP, en la siguiente sección se amplía el tema.

1.3 Metaheurísticas. Algoritmos Evolutivos

Dentro de los AE o Computación Evolutiva como también se denominan, se enmarcan los AG como los más célebres y conocidos representantes de las Metaheurísticas. Por ello se hace necesario abordar brevemente el contenido (9, pág. 16).

1.3.1 Metaheurísticas

El término meta-heurísticas se obtiene de anteponer a heurística el sufijo meta que significa un nivel superior. Apareció por primera vez en el artículo seminal sobre búsqueda tabú de Fred Glover en 1986 (5, pág. 15). Trata básicamente de combinar los métodos heurísticos básicos en marcos de trabajo del más alto nivel lanzándose a la exploración de un espacio de búsqueda. Desarrollado en la actualidad, como método para abordar la solución de los problemas de optimización combinatoria con mayores requerimientos computacionales (9, pág. 45).

Propiedades fundamentales que caracterizan a las Metaheurísticas (9, pág. 10-11):

- Son estrategias que guían el proceso de búsqueda.
- La meta es explorar eficientemente el espacio de búsqueda para encontrar soluciones (sub) óptimas.
- Las técnicas que constituyen los algoritmos catalogados como Metaheurísticas van desde procedimientos simples de búsqueda local a complejos procesos de aprendizaje.
- Son algoritmos aproximados y no determinísticos.
- Incorporan mecanismos para evitar quedarse atrapados en áreas prometedoras cerradas del espacio de búsqueda.
- Los conceptos básicos asociados a las Metaheurísticas permiten una descripción de nivel abstracto del espacio de búsqueda y la evolución del mismo.
- No son específicas del problema.

- Hacen uso de conocimiento específico del dominio y/o experiencia de búsqueda (memoria) para sesgar la búsqueda.

En resumen, las Metaheurísticas son conceptos de alto nivel para explorar espacios de búsqueda usando diferentes estrategias. Estas deben ser escogidas de tal forma que exista un equilibrio dinámico entre la explotación de la experiencia de búsqueda acumulada (intensificación o explotación) y la exploración del espacio de búsqueda (diversificación). Este equilibrio es necesario por una parte, para identificar rápidamente regiones en el espacio de búsqueda con soluciones de alta calidad, y por otra, para no invertir demasiado tiempo en regiones del espacio de búsqueda que han sido ya exploradas, o no proporcionan soluciones de alta calidad (9, pág. 11).

Entre los algoritmos más representativos (9, pág. 10) y comúnmente conocidos que incluye están:

- Inteligencia enjambre
- Algoritmos Evolutivos
- Búsqueda local Iterativa
- Enfriamiento simulado

Este trabajo se centra en los AE, de ahí que a continuación, se presente una introducción de los mismos.

1.3.2 Algoritmos Evolutivos

Son algoritmos de búsqueda estocásticos que reproducen la evolución biológica natural. Todos los AE se basan en el concepto de una población de individuos que son representados por puntos de búsqueda en el espacio de soluciones, los cuales, empleando operadores probabilísticos de mutación, selección y en ocasiones recombinación, evolucionan hasta encontrar los mejores individuos. El costo de un individuo se refleja con respecto a una función objetiva particular que se desea optimizar. El operador de mutación introduce innovación dentro de la población al añadir alteraciones en los individuos. El operador de recombinación realiza un intercambio de información entre diferentes individuos de la población, mientras que el operador de selección se encarga de escoger a los mejores individuos (6, pág. 33).

Todos los AE, según J. R. Cano en (6, pág. 34), comparten los mismos conceptos básicos, sin embargo, difieren en el mecanismo empleado para codificar las soluciones y los operadores que emplean para producir la siguiente generación. Los parámetros principales a tener en cuenta para cualquiera de estos y ajustar su comportamiento, evitando llegar a un óptimo local como solución, son:

- El tamaño de la población.
- Las tasas de mutación y cruce.

Los valores de mutación se mantienen con probabilidad baja para evitar la introducción de soluciones extrañas, mientras que la probabilidad de cruce es el principal mecanismo para generar nuevos individuos manteniéndose en valores altos.

Las estrategias de selección de los AE se pueden llevar a cabo de diferentes formas, pero en todas ellas se persigue el objetivo de preservar buenos individuos, descartando el resto. Los métodos de selección pueden ser:

- **Suaves:** se asocia una probabilidad de supervivencia a cada individuo, de forma que aquel que ofrezca la mejor solución será el que presente la mayor probabilidad.
- **Estrictos:** se elige un número fijo de las mejores soluciones disponibles (6, pág. 34).

Es importante señalar que los AE del mismo modo que se emplean para la selección de características, permitiendo reducir el tamaño del conjunto original de los datos, pueden ser utilizados para disminuir dicho tamaño seleccionando las instancias más representativas (6).

J. R. Cano plantea en (6, pág. 34) que existen diferentes tipos de AE:

- **Programación genética**

Consiste en la evolución automática de programas usando ideas basadas en la selección natural, permitiendo realizar regresión simbólica para obtener además de un dato numérico predictivo, una expresión matemática en función de las variables de entrada (10). Es básicamente una variante de los AG, diferenciándose fundamentalmente porque en él, los individuos representan programas de ordenador (11).

- **Estrategias de evolución**

Son métodos de optimización paramétricos, que trabajan sobre muestras seleccionadas compuestas por números reales (12).

- **Programación evolutiva**

La Programación Evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación, es decir, diferentes especies no se pueden cruzar entre sí (13, pág. 9).

- **Algoritmos Genéticos (AG)**

Los AG son procedimientos adaptativos para la búsqueda de soluciones de espacios complejos, inspirados en los procesos de evolución natural y evolución genética (9, pág. 16). Estos algoritmos fueron introducidos por Holland para imitar algunos de los mecanismos que se observan en la evolución de las especies. Basándose en estas características, Holland creó un algoritmo que genera nuevas soluciones a partir de la unión de soluciones progenitoras utilizando operadores similares a los de la reproducción, sin necesidad de conocer el tipo de problema a resolver (14, pág. 16).

1.3.3 Tipos de Algoritmos Genéticos

Se pueden distinguir dos modelos dentro de los AG, según E. Yeguas, autor del artículo (9, pág. 21), estos son:

- **Modelo generacional o clásico**

Durante cada generación se crea una población completa con nuevos individuos mediante la selección de padres de la población anterior y la aplicación de los operadores genéticos sobre ellos. La nueva población reemplaza directamente a la antigua.

- **Modelo estacionario**

Durante cada generación se escogen dos padres de la población (usando muestreo aleatorio simple o cualquier otro tipo de muestreo) y se le aplican los operadores genéticos. Para la nueva población se escogen de forma aleatoria los individuos o se reemplazan los individuos más antiguos de la población.

1.3.4 ¿Por qué utilizar Algoritmos Genéticos?

La razón del creciente interés por los AG, expuesta en (5, pág. 24-25), es que estos son un método global y robusto de búsqueda de las soluciones de problemas. La principal ventaja de estas características es el equilibrio alcanzado entre la eficiencia y eficacia para resolver diferentes y muy complejos problemas de grandes dimensiones.

- Trabajan con una codificación de un conjunto de parámetros, no con los parámetros mismos.
- Trabajan con un conjunto de puntos, no con un único punto y su entorno (su técnica de búsqueda es global.) Utilizan un subconjunto del espacio total, para obtener información sobre el universo de búsqueda, a través de las evaluaciones de la función a optimizar. Esas evaluaciones se emplean de forma eficiente para clasificar los subconjuntos de acuerdo con su idoneidad.
- No necesitan conocimientos específicos sobre el problema a resolver; es decir, no están sujetos a restricciones. Por ejemplo, se pueden aplicar a funciones no continuas, lo cual les abre un amplio campo de aplicaciones que no podrían ser tratadas por los métodos tradicionales.
- Utilizan operadores probabilísticos, en vez de los típicos operadores determinísticos de las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.
- Cuando se usan para problemas de optimización, resultan menos afectados por los máximos locales que las técnicas tradicionales (son métodos robustos).

1.3.5 Ventajas y desventajas de los Algoritmos Genéticos

Los AG están ampliamente difundidos por su versatilidad para la resolución de problemas de optimización y son aplicados a un amplio rango de problemas pertenecientes a campos tan diversos como la robótica, ingeniería, inteligencia artificial o economía debido a que presenta las siguientes características (5, pág. 23-24):

- Pueden resolver problemas difíciles de forma rápida y fiable.
- Aprenden a mantener o eliminar posibles soluciones en función de su calidad.

- Son ciegos, en el sentido de que no manejen ningún tipo de información sobre el problema en concreto, exceptuando la función de evaluación.
- Aunque no hay garantía de encontrar la solución óptima, generalmente encuentran soluciones aceptables.
- Se pueden hibridar fácilmente.
- Alta capacidad para explotar la información acumulada sobre un espacio de búsqueda y, de este modo, dirigir las siguientes búsquedas hacia los mejores subespacios. Por esto se aplican sobre espacios grandes, complejos y parcialmente definidos donde las técnicas clásicas de búsqueda no son apropiadas.

Sin embargo, existen desventajas (5, pág. 24) asociadas a los AG, estas son:

- La convergencia prematura hacia zonas del espacio de búsqueda que no contienen el óptimo global.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida los parámetros que se utilicen, tamaño de la población, número de generaciones, etc.

1.3.6 Aplicaciones de los Algoritmos Genéticos

Entre las muchas aplicaciones (5, pág. 25) que presentan los AG está su utilidad en:

- Control de procesos químicos.
- Optimización estructural.
- Optimización combinatoria y en dominios reales.
- Modelado e identificación de sistemas.
- Planificación y control.
- Ingeniería.
- Vida artificial.
- Planificación de sistemas de Producción.
- El aprendizaje, la Clasificación y la minería de datos.
- Internet y los Sistemas de Recuperación de Información.

1.3.7 Algoritmos Evolutivos aplicados a la Selección de Instancias

- **Algoritmo Genético Generacional (AGG)**

Este modelo, también conocido como Algoritmo Genético Canónico o Simple (15, pág. 34) es el modelo clásico del algoritmo genético. Consiste en las siguientes operaciones:

1. Generar una población inicial de soluciones.
2. Seleccionar, de la población actual, las soluciones mejor adaptadas.
3. Cruzar algunas soluciones para obtener su descendencia.
4. Mutar algunas soluciones para obtener las soluciones mutadas.
5. Elegir las soluciones que sobreviven y formarán la nueva generación.
6. Si no se alcanza el criterio de parada volver al paso 2.

Al finalizar los pasos anteriores, la mejor solución de la población es la que se propone como solución del problema.

- **Algoritmo Genético Estacionario (AGE)**

El modelo de evolución de estado estacionario consiste en una interesante propuesta para mantener el equilibrio entre los mecanismos de exploración y de explotación del algoritmo evolutivo (16, pág. 6). En este algoritmo cada paso de evolución consiste en (16, pág. 4-5) (6, pág. 50):

- La selección de dos padres para efectuar un cruzamiento, produciendo uno o eventualmente dos descendientes.
- El o los individuos resultantes pueden ser mutados probabilísticamente.
- El o los individuos generados se insertan en la población, reemplazando algunos individuos preexistentes.
- El criterio de reemplazo puede ser aleatorio, elitista o proporcional al *fitness*, es decir, puede ser el valor de una función objetivo, el resultado de un experimento de simulación o alguna otra clase de medida.
- Normalmente, se trabaja en hipótesis de competencia entre el descendiente generado y el individuo a reemplazar.

El AGE se diferencia del AGG en que para cada miembro de la población creada, el modelo de estado estacionario necesita realizar el doble de selecciones. El algoritmo de estado estacionario es más rápido para encontrar mejores soluciones que el algoritmo generacional, no obstante, sus soluciones pueden ser superadas a largo plazo por el algoritmo generacional que cuenta con un patrón de exploración de espacio de búsqueda superior (16, pág. 8).

- **PBIL**

Con el objetivo de disminuir las desventajas de los operadores de recombinación usuales de los algoritmos de Computación Evolutiva, se desarrollaron una serie de algoritmos llamados Algoritmos de Estimulación de Distribución (EDAs).

Estos algoritmos se basan en la teoría de la probabilidad y en las poblaciones que evolucionan a medida que lo hace el proceso de búsqueda. Los EDAs usan modelado probabilístico de soluciones prometedoras para estimular una distribución sobre el espacio de búsqueda. Tal distribución se usa entonces para crear la siguiente generación al muestrear el espacio de búsqueda según la misma. Después de cada iteración se reestima la distribución (9, pág. 60).

PBIL se considera el modelo inicial de los algoritmos de evolución basados en estimaciones de distribuciones, es una combinación de algoritmo genético y aprendizaje competitivo diseñado para búsquedas en espacios binarios. El algoritmo PBIL mantiene explícitamente estadísticas sobre el espacio de búsqueda para decidir cuál es el siguiente conjunto a muestrear (6, pág. 51-52).

- **CHC**

Este algoritmo (6, pág. 50-51) es un método reportado en la literatura especializada de AG, siendo una de las primeras propuestas en introducir mecanismos de diversidad para alcanzar un buen equilibrio entre explotación y exploración en el proceso de búsqueda. Sigue las siguientes operaciones:

  - Selecciona una población de padres de tamaño N.
  - Generar una población intermedia de N individuos.
  - Se emparejan aleatoriamente y se emplean para generar los N potenciales descendientes.
  - Conformar la siguiente generación seleccionando las N mejores muestras de entre las poblaciones de padres e hijos.

Los resultados de la investigación mostraron que los algoritmos más prometedores son PBIL y CHC. Ofrecen el mejor comportamiento en selección de conjuntos de entrenamiento, independientemente del tamaño del conjunto de datos considerado, así como los mejores porcentajes de reducción junto con las mayores tasas de precisión. El CHC se destaca por presentar el menor costo computacional de entre todos los AE y probabilísticos (6, pág. 70). Por lo anteriormente planteado se seleccionan el PBIL y el CHC como algoritmos para la reducción de instancias en las muestras de la plataforma alasGRATO.

1.4 Programas vinculados a la Selección de Variables en el mundo

- **Keel**

Es un *software* para evaluar la evolución de los algoritmos de minería de datos y problemas de regresión, entre ellos: clasificación, agrupamiento, patrón de la minería. Contiene una gran colección de algoritmos clásicos de extracción de conocimientos, técnicas de pre procesamiento (SI, selección de características, discretización, métodos de imputación de valores), Inteligencia Computacional de aprendizaje basado en algoritmos; incluido el estado evolutivo de algoritmos de aprendizaje basados en diferentes enfoques (Pittsburgh, Michigan e IRL) y modelos híbridos como sistemas difusos genéticos, redes neuronales evolutivas. Permite realizar un análisis completo de cualquier modelo de aprendizaje en comparación con los existentes, incluido un módulo de prueba estadística para la comparación entre ellos. El uso más común de esta herramienta para un investigador será la ejecución automatizada de los experimentos y el análisis estadístico de sus resultados. No está diseñada para ofrecer un tiempo real del progreso de los algoritmos. Trabaja muy bien en ambiente distribuido de sistemas. Fue diseñado con doble objetivo: la investigación y la educación. Cuenta con licencia comercial, lo que lo convierte en s*oftware* propietario además, su versión 1.0 es de código abierto (14, pág. 27-28).

- **RapidMiner (anteriormente Yale)**

Es un *software* de código abierto para el análisis inteligente de datos, descubrimiento de conocimientos, minería de datos, aprendizaje automático, visualización; con numerosas características y funciones para la selección de variables. Constituye además, un entorno de aprendizaje automático y de extracción de datos para todo tipo de experimentos. Permite que los experimentos sean realizados con un gran número de variables arbitrarias, las cuales se escriben en archivos XML que son fácilmente creados con la interfaz gráfica de RapidMiner. Ofrece más de 400 operadores para los principales procedimientos

de aprendizaje de máquinas, incluidos los de entrada, salida, pre procesamiento de datos y visualización de los mismos. Está escrito en el lenguaje de programación Java y por tanto pueden trabajar en todos los Sistemas Operativos populares. También integra todos los sistemas de aprendizaje y de atributo de los evaluadores Weka. Cuenta con una licencia GNU GPL, Propietaria y Comercial (14, pág. 27).

- **Weka**

Es un paquete de *software* de Java para la extracción de conocimientos desde bases de datos; incluye además, una recopilación de algoritmos de aprendizaje automático para tareas de minería de datos. Este *software* ha sido desarrollado en la universidad de Waikato (Nueva Zelanda) bajo la licencia GPL, lo que significa que este programa es de libre distribución y difusión; lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años. Es de gran utilidad al ser utilizado mediante las interfaces que ofrece o para embeberlo dentro de cualquier aplicación. Además Weka contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, *clustering*, asociación y visualización. Está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla. Sin embargo, tiene un gran defecto y es la escasa documentación orientada al usuario que tiene junto a una usabilidad bastante pobre, lo que la hace una herramienta difícil de comprender y manejar sin información adicional. Además, como Weka está programado en Java, es independiente de la arquitectura, ya que funciona en cualquier plataforma sobre la que haya una máquina virtual de Java disponible. Una de las propiedades más interesantes de este *software*, es su facilidad para añadir extensiones y modificar sus métodos (14, pág. 28).

1.5 Conclusiones Parciales

Luego de realizada una minuciosa investigación sobre la SI se arriba a las siguientes conclusiones:

- La SP, estrategia mediante la cual se puede llevar a cabo la selección de conjuntos de entrenamiento, es la técnica más prometedora dentro de la SI para obtener modelos predictivos de menor tamaño y mayor interpretabilidad.
- La Selección basada en AE ofrece simultáneamente dos ventajas: mayor reducción del conjunto de datos y precisión elevada.

- Los AE más destacados aplicados a la SI son PBIL y CHC. Ambos permiten escoger las muestras más representativas del conjunto independientemente de su posición en el espacio de búsqueda.

Finalmente, tras el estudio y análisis efectuado se exponen algunas de las herramientas vinculadas a la SI en el mundo.

CAPÍTULO 2

MATERIALES Y MÉTODOS

Se presentan en la sección de Métodos, una breve descripción de las características de las muestras, los AE para la reducción del espacio muestral de forma vertical y se explican las peculiaridades de los mismos. Se definen las herramientas y el entorno de desarrollo que se utilizaron para la implementación y evaluación de los algoritmos seleccionados.

Métodos

2.1 Características de las Muestras

2.1.1 Cefalosporinas

Las cefalosporinas son compuestos antibacteriales pertenecientes a la familia de los β-lactámicos. Todas las cefalosporinas se derivan de la cefalosporina C, un antibiótico natural producido por la cepa de Cephalosporium acremonium aislado por primera vez en 1945. Las cefalosporinas tienen un anillo β–lactámico y presentan un anillo de dihidrotiazina, como se muestra en la Figura 5 (14, pág. 30).

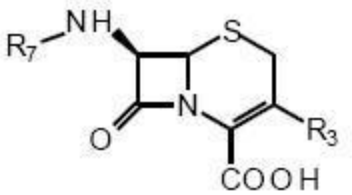


Figura 2.1: Esquema general de la estructura de cefalosporinas

Las cefalosporinas poseen un amplio rango de actividad antibacterial, una excelente tolerancia en niños y casi ninguna toxicidad asociada a las dosis. Estos antibióticos pueden emplearse con seguridad en niños de todas las edades con fallos renales o hepáticos (14, pág. 30).

La muestra empleada consta de 104 compuestos pertenecientes a cuatro generaciones distintas de cefalosporinas.

2.1.2 Inhibidores del Factor Esteroidogénico-1 (Ensayo_599)

El receptor nuclear SF-1 (factor esteroidogenico-1) pertenece a la clase de receptores nucleares huérfanos que han sido poco investigados al nivel farmacológico (celular) que se reporta[2].

El SF-1 se expresa en las glándulas adrenal, pituitaria, testículos, y ovarios y regula la producción de la hormona esteroidea a diferentes niveles, incluyendo la expresión directa de la enzima P-450 principal involucrada en la síntesis de dicha hormona (14, pág. 31).

Este es un ensayo de dosis-respuesta tipo confirmatorio basado en células para la inhibición del receptor huérfano. La muestra, estructuralmente heterogénea, está formada por 315 instancias.

2.2 Algoritmos Genéticos

Los AG son algoritmos matemáticos altamente paralelos que transforman un conjunto de objetos matemáticos individuales con respecto al tiempo. Estos usan operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud (14, pág. 16-18). Para la aplicación de este algoritmo deben definirse ciertos aspectos fundamentales:

- Cómo inicializar una población para garantizar la mayor diversidad de soluciones y que el procedimiento no converja prematuramente. En este sentido, se tienen dos variantes de solución, una es inicializar aleatoriamente toda la población, y la otra es sembrar (fijar) un individuo (conjunto de datos) en la población inicial (muestra), para acelerar el proceso evolutivo (optimización).
- La forma de evaluar un individuo. Existen sin embargo, dos problemas importantes asociados a este método, como son la competición próxima (individuos cuya aptitud relativa son próximas numéricamente, están numéricamente agrupados) y el efecto de súper individuos, individuos con evaluación muy superior a la media, capaces de dominar el

[2] Ensayo 599 de la base de datos del *National Center for Biotechnology Information* por sus siglas NCBI http://www.ncbi.nlm.nih.gov/Entrez/

proceso de selección, haciendo que el AG converja prematuramente hacia un óptimo local (posible outlier).

- Los operadores genéticos que proporcionen un mecanismo estructurado de intercambio de información útil (bloques constructivos para el cruzamiento) entre individuos y a su vez explorar nuevas zonas del espacio de búsqueda que permita escapar de extremos locales.
- Cómo seleccionar los individuos (soluciones) para la próxima generación de modo que contribuya geométricamente a la presencia de esquemas aventajados y reducir la presencia de los retrasados. Dada la variedad existente de esquemas de selección, se desarrolla un mecanismo de selección elitista combinado con selección por ruleta, donde las dos mejores soluciones de la población actual son insertadas en la siguiente generación para mejorar el proceso de convergencia del algoritmo, y los restantes individuos se seleccionan probabilísticamente en proporción a la aptitud que estos posean.
- La condición de parada, que puede ser por un número fijo de generaciones o cuando el algoritmo converge a una misma solución, lo cual se explica cuando toda la población posea una misma solución.

Los AG constituyen una técnica robusta que posibilita tratar exitosamente disimiles problemas procedentes de diferentes campos, incluso en aquellos donde otros métodos presentan dificultades. Son utilizados para la SI, demostrando un excelente comportamiento como selector evolutivo de prototipos y garantizando soluciones de un nivel aceptable en un tiempo admisible si no encuentra la solución óptima del problema (17, pág. 2).

A continuación se exponen los AE seleccionados para la reducción de instancias como solución al problema existente en la plataforma alasGRATO.

2.3 CHC

El algoritmo CHC es una variante del AG clásico. Utiliza una estrategia de selección elitista y el operador de cruzamiento uniforme (HUX), que intercambia exactamente la mitad de la información diferente entre dos individuos padre. Solo se permite el cruce entre individuos cuya información difieran una cierta distancia, inicializada en 1/4 del largo del elemento utilizado y disminuida en 1 en cada generación en que no se generan descendientes. La diversidad no se introduce mediante un operador de

mutación, sino por un mecanismo de reinicialización, que se aplica al detectar la convergencia (9, pág. 50).

2.3.1 Estructura del algoritmo

El algoritmo CHC sigue los siguientes puntos en su estructura, y difiere del AG tradicional en todos menos en el primero (9, pág. 50-51):

1. La inicialización de la población P (0) es aleatoria.
2. La selección para la supervivencia (selección elitista) está orientada hacia la selección de las mejores estructuras.
3. Se introduce un nuevo sesgo en contra de los individuos emparejados que son similares (prevención de incesto).
4. El operador de recombinación, es una variante del cruce uniforme (HUX).
5. La mutación no se ejecuta en la etapa de recombinación, y se mantiene la diversidad (o más precisamente reintroducida) por aleatorización parcial en la población siempre que se detecta convergencia (reinicialización).

CHC introduce cuatro componentes (9, pág. 51-52):

- **Selección Elitista**

Selecciona los PS mejores elementos entre padres e hijos. Los PS mejores encontrados hasta el momento permanecerán en la población actual.

- **Cruce Uniforme HUX**

Intercambia exactamente la mitad de los posibles estados del elemento que son distintos en los padres. Garantiza que los hijos tengan una distancia Hamming máxima a sus dos padres.

- **Prevención de Incesto**

Se forman PS/2 parejas con los elementos de la población. Sólo se cruzan las parejas cuyos miembros difieren en un número determinado de bits (umbral de cruce). El umbral se inicializa a la L/4 (L es la Longitud del elemento). Si durante el siclo no se produce ni un solo cruce, al umbral de cruce se le resta uno.

- **Reinicialización**

Cuando el umbral de cruce es menor que cero, la población se reinicializa, usando el mejor elemento como plantilla e incluyendo una copia suya, o manteniendo el mejor o parte de los mejores de la población y el resto aleatorio.

A continuación se muestra el pseudocódigo del algoritmo CHC implementado en la aplicación:

```
Procedimiento
t = 0;
Inicializar Poblacion P(t);
Escalar P(t);
Evaluar P(t);
    Hacer
        t = t + 1;
        SeleccionarParaReproduccion C(t) de P(t-1);
        Recombinar estructuras en C(t) para formar C'(t);
        Evaluar estructuras en C'(t);
        Si EsIgual (P(t-1), P(t)) entonces
            DecrementarUmbralDiferencia (D);
        FinSi
        Sino
            SeleccionarParaSupervivencia P(t) de C'(t) y P(t-1);
        FinSino
        Si (D < 0) entonces
            Reinicializar;
        FinSi
    Mientras (no se cumplan las condiciones de parada);
FinProcedimiento
```

Figura 2.2: Pseudocódigo del algoritmo CHC

2.3.2 Selección Elitista

Durante la selección para la reproducción, en vez de sesgar la selección de los candidatos C(t) para la reproducción en favor de los miembros de mejor rendimiento de la población padre P(t - 1), cada miembro de P(t - 1) se le asigna a C(t), y se equilibra aleatoriamente para la reproducción. Durante la selección para la supervivencia, por otra parte, en vez de reemplazar la vieja población padre P (t - 1) con la población hija C(t) para formar P(t), los hijos recién creados deben competir con los miembros de la población padre P(t - 1) para la supervivencia. Los miembros de P (t - 1) y C (t) se mezclan y ordenan de acuerdo a su *fitness* o rendimiento, y P (t) se crea seleccionando los mejores PS miembros (donde PS es el tamaño de la población) de la población mezclada. En casos en los que un miembro de P (t - 1) y un miembro de C (t) tienen el mismo *fitness*, el miembro de P (t - 1) se dispone en la posición más alta del *ranking*. Conocemos a este procedimiento de retener los mejores miembros ordenados de las poblaciones padre e hija mezcladas como selección elitista de la población, ya que garantiza que los mejores PS individuos generados siempre sobrevivirán.

```
Procedimiento SeleccionarParaReproduccion
    Copiar todos los miembros de P(t-1) en C(t) aleatoriamente
FinProcedimiento
```

Figura 2.3 Estructura de las funciones de selección para la reproducción

```
Procedimiento SeleccionarParaSupervivencia
    Obtener P(t) a partir de P(t-1) con los mejores miembros de C'(t)
    hasta que no queden miembros de C'(t)
    que sean mejores que ningún miembro restante de P(t-1)
FinProcedimiento
```

Figura 2.4: Estructura de las funciones de selección para la supervivencia (Selección Elitista)

2.3.3 Cruce Uniforme HUX

El operador de recombinación usado por CHC es una variante del cruce uniforme, intercambia bits en vez de segmentos. Para cada posición en la cadena, los bits de los dos padres se intercambian con probabilidad fija p (típicamente 0.5). Este operador cruza sobre exactamente la mitad de los genes diferentes, donde los bits que se intercambiarán se eligen aleatoriamente sin reemplazamiento. Disminuyendo el peligro de la convergencia prematura y maximizando la oportunidad de dos buenos

esquemas, uno por cada padre, quedando combinados en un hijo, puesto que se escoge la mitad del material de cada padre.

```
Procedimiento Recombinacion
    Para (cada una de las PS/2 parejas de estructuras de C(t)) hacer
        Intercambiar la mitad de los bits diferentes aleatoriamente
    FinPara
FinProcedimiento
```

Figura 2.5: Estructura de la función de Cruce HUX

2.3.4 Prevención de Incesto

CHC tiene un mecanismo adicional para retardar la etapa de convergencia, un mecanismo para ayudar a evitar el incesto. Durante la etapa de reproducción, cada miembro de la población padre se elige aleatoriamente sin reemplazamiento y se casa para el emparejamiento. Antes del emparejamiento, se calcula la distancia de Hamming entre padres potenciales, y si la mitad de esa distancia no excede un umbral diferencia, no se emparejan y se borran de la población hija. Se empareja sólo una fracción de la población para producir nueva descendencia en cualquier generación. Siempre que no haya hijos aceptados en la población padre (o bien porque no hay emparejamientos potenciales o bien porque ninguno de los hijos fue mejor que el peor miembro de la población padre), se decrementa el umbral de diferencia.

La prevención de incesto se incorpora por tanto en la función de cruce del algoritmo de evolución CHC, modificando su discurrir en la forma indicada. La función definitiva que corresponde al cruce dentro del algoritmo CHC queda en pseudocódigo como se presenta en la figura 10.

```
Procedimiento Recombinar
    Para (cada una de las PS/2 parejas de estructuras de C(t)) hacer
        Determinar la distancia de Hamming
        Si (distanciaHamming/2) > D entonces
            Intercambiar la mitad de los bits diferentes aleatoriamente
        Sino
            Borrar la pareja de estructuras de C(t)
        FinSi
    FinPara
FinProcedimiento
```

Figura 2.6: Estructura de la función definitiva para el Cruce HUX

2.3.5 Reinicialización

CHC introduce la mutación sólo cuando la población ha convergido o la búsqueda se ha estancado. Más específicamente, siempre que el ciclo reproducción-recombinación logra su condición de terminación, la población se reinicializa (diverge) y se repite el ciclo. La reinicialización, sin embargo, es sólo parcial. La población se reinicializa usando el mejor individuo encontrado como plantilla para crear una nueva población. Se crea cada nuevo individuo alterando una proporción fija.

Así mismo, se añade una instancia de los mejores individuos sin cambiar. Esto asegura que la siguiente búsqueda no pueda converger a una solución peor que la previa. Finalmente, el umbral diferencia se reinicializa como indicador de convergencia a partir del ratio de divergencia y teniendo en cuenta la longitud de los cromosomas. Este bucle más externo, es iterado hasta que se alcanza la condición de terminación.

La ventaja que aportan las reinicializaciones parciales sobre la mutación, es que CHC puede trabajar bastante bien en un amplio rango de problemas usando la misma configuración de parámetros.

```
Procedimiento Reinicializar
    Reemplazar M individuos en P(t) con mejores M de P(t-1)
    Para todos excepto los mejores M de P(t) hacer
        Reemplazar el individuo con el mejor global
        Cambiar DR * L bits aleatoriamente
    FinPara
    EvaluarPoblacion (P(t))
FinProcedimiento
```

Figura 2.7: Estructura de la función de reinicialización

2.4 PBIL

E. Yeguas expone en (9, pág. 60) que el algoritmo PBIL intenta crear un vector de probabilidades del que se obtienen muestras o individuos representativos en cada iteración para producir la población que constituye la generación correspondiente. Tal y como ocurre en el AG tradicional, se asume que cada solución o individuo se codifica a partir de un vector de longitud fija. La población se reemplaza por un simple vector de probabilidades, que especifica, en cada posición, la probabilidad de contener un valor especifico. Ignorando la contribución del resto de operadores en cada generación, la distribución de los

valores esperada en cada posición de la población durante una generación determinada, puede calcularse sobre la base de la población generada en la generación anterior.

A partir de un mismo vector de probabilidades se puede generar una gran multiplicidad de poblaciones diferentes, dependiendo de la etapa de búsqueda. Pero, los individuos generados por PBIL corresponden generalmente a una determinada región del espacio de búsqueda, delimitada por el vector de probabilidades (9, pág. 64).

E. Yeguas (9, pág. 62-63) plantea además que a pesar de que el algoritmo PBIL utilice el vector de probabilidades para definir una población, no realiza sin embargo, el proceso contario, es decir, generar a través de la población el vector de probabilidades.

Tiene dos formas de definir un operador de mutación (9, pág. 65): la primera es utilizar el operador de mutación directamente sobre los vectores generados y la segunda es utilizarlo sobre el vector de probabilidades. Ambos tipos de mutación tienen el mismo efecto: ayudar a preservar la diversidad. La mutación tiene su mayor importancia en las etapas tardías de la búsqueda, cuando se pierde la diversidad en la población. El cruce es válido en fases tempranas de la búsqueda en la medida en que supone pasos más amplios a la hora de encontrar buenas soluciones. El ajuste final de las soluciones lo realiza el operador de mutación.

La regla de mutación correspondiente a PBIL se muestra a continuación en la Figura 12.

```
Procedimiento Mutacion

  Entradas
  PV: vector de probabilidades
  MP: probabilidad de mutación
  MS: desplazamiento de mutación

  Para (cada valor del vector de probabilidades) hacer
     Si (aleatorio [0,1] < PM) entonces
        PV (i)=PV (i)*(1.0-MS)+aleatorio (0.0 o 1.0)*MS
     FinSi
  FinPara

FinProcedimiento
```

Figura 2.8: Estructura genérica más utilizada para la mutación en PBIL

El aprendizaje competitivo, es usado a menudo para clasificar una serie de muestras no etiquetadas en distintos grupos. La pertenencia a cada grupo se basa en la similitud de las muestras con respecto a las características en estudio. El objetivo es que el aprendizaje competitivo sea capaz de

determinar las características más relevantes, así como de clasificar en base a las mismas. La regla de actualización del vector de probabilidades, en el algoritmo PBIL básico, queda, tomando como punto de partida la ecuación:

$$pi = pi\_ant \times (1{,}0 - LR) + mejori \times LR$$

Donde pi es el valor i-ésimo del vector de probabilidades que se actualiza, pi_ant es el valor i-ésimo que el vector de probabilidades tenía previo a la actualización, LR es el ratio de aprendizaje y mejori es el valor i-ésimo de la mejor solución encontrada en la generación actual. Puesto que en PBIL, el vector de probabilidades se usa para generar el siguiente conjunto de soluciones, el ratio de aprendizaje también afecta a qué regiones del espacio de búsqueda son exploradas. El valor del ratio de aprendizaje tiene un impacto directo en el equilibrio entre exploración y explotación del espacio de búsqueda (cuanto más alto es el ratio de aprendizaje mayor es la explotación y cuanto más bajo mayor es la exploración).

Tal y como ocurría en CHC, en PBIL podemos introducir reinicialización para combatir la convergencia del algoritmo, que, aunque se puede retardar con el ajuste del parámetro correspondiente al ratio de aprendizaje, es bastante rápida. Así, una vez que se comprueba que el vector de probabilidades ha convergido, reinicializaremos el vector de probabilidades a su forma inicial (valores a 0.5), almacenando, en todo caso, la mejor solución encontrada hasta ese momento. Véase la Figura 13.

```
Procedimiento Reinicializar

  Si Estancamiento (PV) entonces
    Para todas las posiciones de PV hacer
      PV (i)=0.5
    FinPara
  FinSi

FinProcedimiento
```

Figura 2.9: Reinicialización del vector de probabilidades

A continuación se presenta el pseudocódigo implementado en la aplicación para el algoritmo PBIL:

```
Procedimiento
    t = 0;
    Inicializar Vector de Probabilidades (PV);
    Inicializar Poblacion P(t);
    mejor = P(0);
    Evaluar P(t);
    Para (i = 1, ... , P(t)) hacer
        Encontrar Mejor Solucion;
    FinPara
    Actualizar vector de probabilidades;
    Si (numAletarorio < probMutacion) entonces
        Realizar Mutacion;
    Fin Si
    Mientras (no se cumplan las condiciones de parada) hacer
        Actualizar Mejor Global;
        Mientras (no se cumplan las condiciones de parada) hacer
            Actualizar vector de probabilidades;
            Si (numAletarorio < probMutacion) entonces
                Realizar Mutacion;
            Fin Si
        FinMientras
    FinMientras
FinProcedimiento
```

Figura 2.10: Pseudocódigo del algoritmo PBIL

Materiales

2.5 Metodologías y herramientas para el desarrollo del sistema

Las metodologías y herramientas utilizadas para el desarrollo de la aplicación que se implementará como parte de la investigación son las establecidas en las pautas de arquitectura de la plataforma alasGRATO, garantizando así, que todo lo desarrollado en la investigación esté conforme con lo realizado

en el proyecto y no presente contradicciones futuras. La plataforma utiliza Java como lenguaje de programación, Eclipse en su versión 3.4 como entorno de desarrollo y la Herramienta CASE (*Computer Aided Software Engineering*), *Visual Paradigm* para el modelado de diagramas. Adoptó OpenUP como metodología de desarrollo.

2.5.1 Plataforma de Desarrollo y Lenguaje de Programación

La plataforma escogida para el desarrollo de la herramienta fue jdk versión 1.5.0_10. Se optó por Java como lenguaje de programación. El mismo fue desarrollado a principio de los años 90´s por *Sun Microsystems*. Oak, como primero se le llamó al lenguaje Java, es semejante en su sintaxis con *C, C++ y Objetive C,* pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria (18).

Entre sus características principales se encuentran:

- Es un lenguaje orientado a objetos.
- Multiplataforma
- Seguro
- Incluye soporte para la comunicación con equipos mediante red, el acceso a bases de datos, creación de páginas HTML dinámicas y aplicaciones visuales al estilo Windows.
- Es fácil de aprender y de usar.

2.5.2 Entorno de desarrollo

El entorno de desarrollo utilizado fue Eclipse en su versión 3.4, es una plataforma de programación válida para crear entornos integrados de desarrollo (IDE, *Integrated Development Environment*). Fue desarrollado originalmente por IBM (*Internacional Bussines Machines*) pero actualmente es desarrollado por la Fundación Eclipse, organización que promueve una comunidad de código abierto. En sí mismo eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (*plug-in*). Este entorno de desarrollo integrado soporta varios lenguajes de programación, sin embargo, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. Esta plataforma, además, facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo (19).

2.5.3 Herramienta CASE

La Herramienta CASE *Visual Paradigm* utiliza el lenguaje unificado de modelado (UML*, Unified Modeling Lenguage*). Tiene disponible las siguientes versiones: *Enterprise, Professional, Standard, Modeler, Personal* y *Community* (que es gratuita). Soporta el ciclo de vida completo del desarrollo de un *software*, desde la fase de análisis hasta el despliegue del mismo. *Visual Paradigm* es una herramienta libre que permite el modelado de varios lenguajes de programación y otras tecnologías de forma fácil y asequible. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Entre sus más recientes características se incluyen el modelado colaborativo con el sistema de control de versiones (CVS, *Concurrent Versions System*) y Subversion, así como la interoperabilidad con modelos UML2 (meta modelos UML 2.x para plataforma Eclipse) a través de XML de intercambio de metadatos (*XMI o XML Metadata Interchange*) (20).

2.5.4 Metodología OpenUP

OpenUP es una metodología open source desarrollada por la IBM y que forma parte del Eclipse Foundation. Es derivada de RUP (*Rational Unified Process*), pero fue simplificada para ser transformada en una metodología más ágil para proyectos de desarrollo de *software*. Es un proceso interactivo de desarrollo de *software* simplificado, completo y extensible, dirigido a gestión y desarrollo de proyectos de *software* basados en desarrollo iterativo, ágil e incremental; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo (21).

OpenUP se caracteriza (22) por cuatro principios básicos que se soportan mutuamente:

- **Colaboración** para alinear los intereses y un entendimiento compartido.
- **Balance** para confrontar las prioridades, es decir, necesidades y costos técnicos para maximizar el valor para los *stakeholders*.
- **Enfoque** en articular la arquitectura para facilitar la colaboración técnica, reducir los riesgos y minimizar excesos y trabajo extra.
- **Evolución** continua para reducir riesgos, demostrar resultados y obtener retroalimentación de los clientes.

Capítulo 2: Materiales y Métodos

Es reconocido mundialmente como uno de los procesos de desarrollo de *software* de mayor calidad, basándose en los principios de Adaptación, Importancia a los involucrados e interesados en los resultados del proyecto; Colaboración, Valor a la iteración y Calidad Continua (21).

CAPÍTULO 3

RESULTADOS Y DISCUSIÓN

En este capítulo se muestra un modelo conceptual para explicar brevemente en qué consiste la aplicación implementada. Se presenta la vista lógica y los patrones de diseño empleados en la programación orientada a objetos. Se describen detalladamente los algoritmos implementados y la validación de los mismos a través de pruebas con datos reales, así como pruebas no paramétricas para comprobar el correcto funcionamiento de la aplicación.

3.1 Modelo de Dominio o Conceptual

El modelo conceptual es un artefacto clásico del análisis orientado a objetos. Explica los conceptos más significativos del problema, identificando los atributos y las asociaciones, es decir, ilustra las interconexiones de los componentes del dominio (23). El modelo conceptual es una descripción del dominio de un problema real, no una descripción del diseño del *software*. Se caracteriza por ser completo, fácil de comprender y explícito en todas sus restricciones. En UML se representa mediante un grupo de diagramas de estructura estática donde no se define ninguna operación. En estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos (atributos) (24).

Las cualidades más significativas del esquema conceptual están orientadas a lograr:

- **Claridad y simplicidad**: significación no ambigua.
- **Coherencia**: ausencia de contradicciones o confusión.
- **Completitud**: sin buscar la exhaustividad, se representa lo esencial de los fenómenos.
- **Fidelidad**: representación sin desviaciones y sin deformaciones.
- **No redundancia**: sólo se representan elementos estrictamente necesarios, y únicamente una vez.

Teniendo en cuenta que no se logra determinar el proceso del negocio con fronteras bien establecidas y ver claramente definidos los actores y trabajadores de cada uno de los procesos. Por estar definido en la metodología OpenUP y por las ventajas que ofrece el diseño de un modelo conceptual para el buen entendimiento de una aplicación orientada a objeto, se decide incluirlo en el presente trabajo.

Los conceptos mostrados en el diagrama son:

- **Usuario**: persona que interactúa con la aplicación.
- **Fichero**: archivo que contiene información sobre los compuestos químicos.
- **Reducción_Instancia**: acción que ejecuta la aplicación para reducir instancias.
- **Fichero_Reducido**: archivo que contiene las muestras reducidas.

En el Modelo de Dominio de la Figura 15, el usuario carga un fichero a partir del cual realiza la reducción de instancias y emite un fichero reducido.

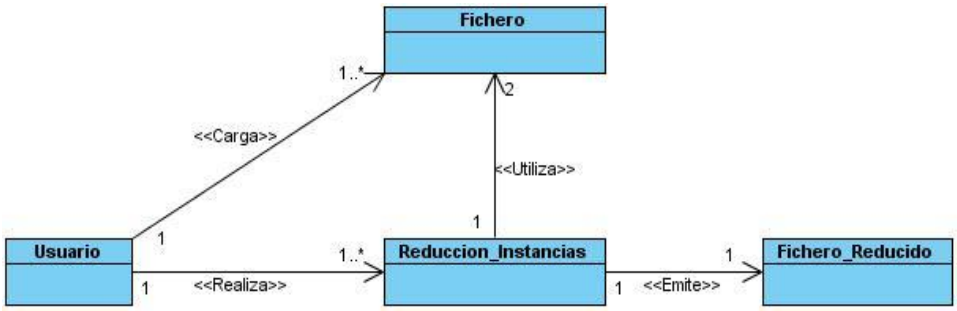


Figura 3.1: Modelo de Dominio de la aplicación implementada

3.2 Diagramas y patrones

Se utiliza el diagrama de vista lógica, porque en él se resumen las clases del diseño. Como los algoritmos se encuentran dentro de los subsistemas, para una mejor compresión del funcionamiento de los mismos, se recomienda ver Anexo No. 1 y Anexo No. 2.

3.2.1 Vista Lógica

La vista lógica permite observar cómo está diseñada la funcionalidad en el interior del sistema. Esta vista describe las clases más importantes, su organización en paquetes de servicio y subsistemas, y la organización de estos subsistemas en capas. Se describen los paquetes de forma abstracta, así como las relaciones que existen entre ellos.

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones, para organizar los distintos componentes (26, pág. 8).

Capítulo 3: Resultados y Discusión

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (26, pág. 8).

- **Modelo**: es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- **Vista**: presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador**: responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

En el patrón MVC evidenciado en la aplicación de la investigación, el componente View contiene la clase Principal, que se relaciona con las clases CenterDesktop para centrar las ventanas y ProgressBar para mostrar una barra de progreso cuando se ejecuta la aplicación. La Principal se relaciona además con la clase SplitSamples, la cual permite al usuario cargar el fichero arff y dividir la muestra en un fichero de entrenamiento y otro de prueba. Según el algoritmo seleccionado, la clase Principal se relaciona con ConfigManagerCHC y ConfigManagerPBIL. Éstas modifican los ficheros configCHC y configPBIL respectivamente, con los datos introducidos por el usuario, así como, la dirección de las muestras de entrenamiento y prueba obtenidas en la clase SplitSamples. La clase Manager controla los subsistemas Librería_CHC y Librería_PBIL para reducción de instancias en dependencia del algoritmo escogido en la ventana Instance Selection. Las muestras reducidas son almacenadas como ficheros arff en el Model.

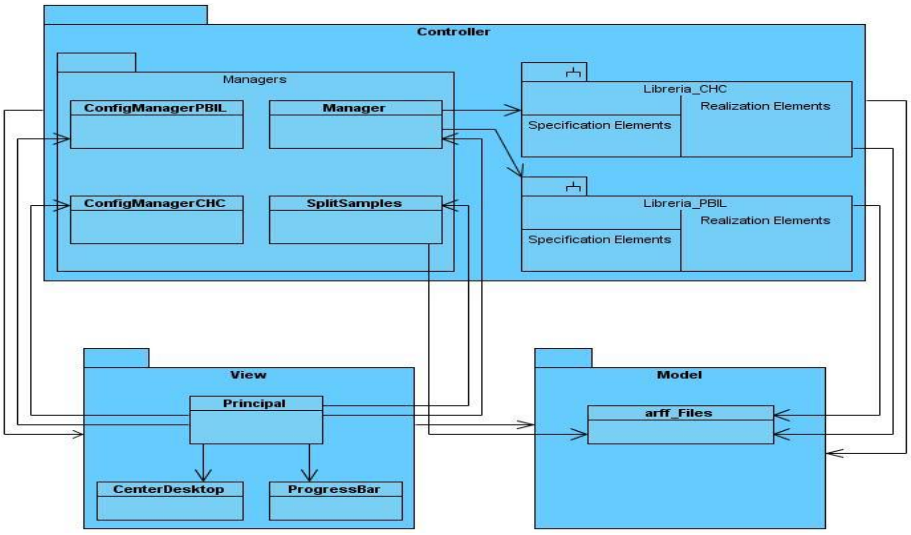


Figura 3.2: Ejemplo de aplicación de patrón MVC

3.2.2 Patrones de diseño utilizados

Los patrones son una pareja de problema / solución, que codifica y estandariza buenos principios relacionados frecuentemente con la asignación de responsabilidades. Son un amplio repertorio de principios generales basados en la experiencia que guían la creación de un *software* (26, pág. 3), así como soluciones simples y factibles a problemas específicos y comunes del diseño orientado a objetos.

3.2.2.1 ¿Qué beneficios produce la utilización de patrones de diseño?

- Contribuyen a reutilizar diseño, identificando aspectos clave de la estructura del diseño que puede ser aplicado en una gran cantidad de situaciones, es decir, provee de numerosas ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de los diseños, proporcionando un considerable ahorro en la inversión.
- Mejoran y elevan la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario.
- Incrementan el vocabulario de diseño, ayudando a diseñar desde un mayor nivel de abstracción.

3.2.2.2 Patrones GRASP

GRASP, patrones generales de *software* para asignar responsabilidades, en inglés *General Responsibility Assignment Software Patterns*. El nombre se deriva de la palabra *grasping* y se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el *software* orientado a objetos (26, pág. 6). Lo patrones de GRASP, no compiten con los patrones de diseño, son una guía para ayudar a encontrar los mismos debido a que son más concretos.

Los principales patrones GRASP son:

- Experto
- Creador
- Alta cohesión
- Bajo Acoplamiento.
- Controlador

Existen además 4 patrones adicionales los cuales son:

- Polimorfismo
- Indirección
- No hables con extraños
- Fabricación Pura

Los utilizados en el trabajo son:

- **Experto**

Tiene como objetivo asignar una responsabilidad al experto en la información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. Con este patrón (27), la encapsulación es mantenida ya que los objetos se valen de su propia información para realizar tareas. Esto permite poco acoplamiento, lo cual conduce a sistemas más robustos y de mantenimiento mucho más fácil. La alta cohesión también es soportada.

En la clase CHC se evidencia el patrón Experto, la misma posee los atributos necesarios para realizar la selección de instancia de acuerdo con el algoritmo CHC. Es decir, a esta clase se le asigna la responsabilidad de ejecutar dicho algoritmo.

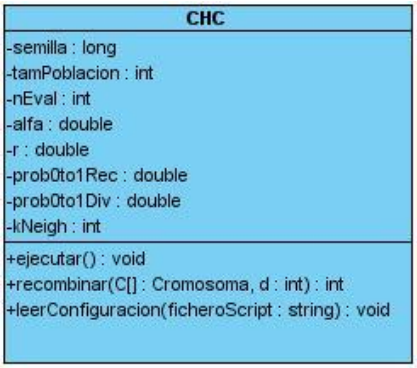


Figura 3.3: Patrón Experto

- **Creador**

Este patrón (28) asigna a la clase B la responsabilidad de crear una instancia de la clase A si alguna de las siguientes premisas es cierta:

  - B agrega los objetos de A

- B contiene los objetos de A
- B registra las instancias de los objetos de A.
- B tiene los datos de inicialización que serán enviados a A cuando este objeto sea creado.

El creador guía la asignación de responsabilidades relacionadas a la creación de objetos, una tarea muy común en sistemas orientados a objetos. Permite que el bajo acoplamiento sea soportado, lo que implica bajo mantenimiento y altas oportunidades de reutilización.

La clase InstanceSet contiene un objeto de la clase InstanceAttributes, de ahí que, sea capaz de asumir la responsabilidad de crear las instancias de la clase InstanceAttribute.

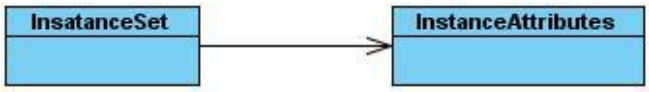


Figura 3.4: Patrón Creador

A continuación se expone el segmento del código que evidencia el Patrón Creador en las clases InstanceSet e InnstanceAttributes.

```
public InstanceSet(InstanceSet is) {
    instanceSet = (Instance[]) Arrays.copyOf(is.instanceSet,
            is.instanceSet.length);
    header = new String(is.header);
    attHeader = new String(is.attHeader);
    attributes = new InstanceAttributes(is.attributes);
    storeAttributesAsNonStatic = is.storeAttributesAsNonStatic;
}
```

Figura 3.5: Patrón Creador (Segmento de código)

- **Alta cohesión**

La alta cohesión es una medida con la que se relacionan las clases y el grado de responsabilidades de un elemento. Una clase tiene responsabilidades moderadas en un área funcional y colabora con otras para realizar las tareas. En la práctica el nivel de cohesión no puede ser considerado independiente de los otros patrones, Experto y Bajo Acoplamiento. Este patrón mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, soportando mayor capacidad de reutilización (28).

En la Figura 20 se refleja este patrón, ya que no se le asignan todas las responsabilidades a la clase Método, sino que, los atributos y métodos necesarios para darle funcionalidad a dicha clase, se

encuentran en las clases Attributes, Attribute, Instance, InstanceSet, KNN, InstanceAttributes. Esto permite un fácil mantenimiento y reutilización de la aplicación.

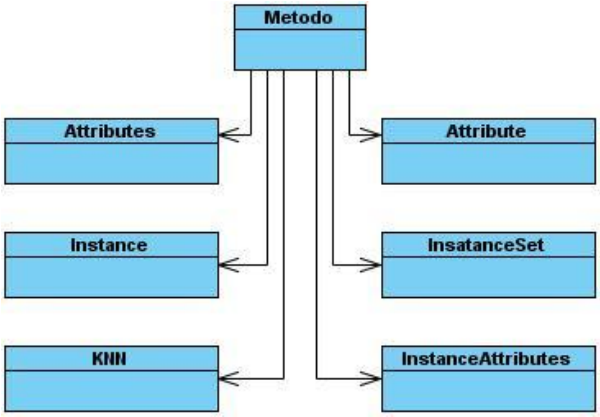


Figura 3.6: Patrón Alta Cohesión

- **Bajo Acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. EL Bajo Acoplamiento significa que una clase no depende de muchas clases. Con el uso de este patrón las clases no se afectan por cambios de otros componentes, son fáciles de entender por separado y de reutilizar (28).

La clase CHC, responsable de ejecutar el algoritmo, utiliza una funcionalidad de la clase Randomize. Esta última se acopla al conocimiento de un objeto de la clase MTwister. De ahí que no se haga necesario para la clase CHC, relacionarse directamente con MTwister para acceder a sus funcionalidades, debido a que su acoplamiento con Randomize le posibilita acceder a la información de esta clase.

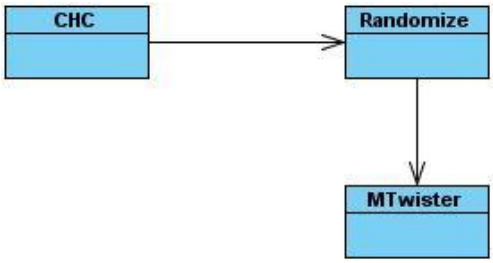


Figura 3.7: Patrón Bajo Acoplamiento

- **Controlador**

Tiene como objetivo asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase que represente alguna de las siguientes opciones (27):

  - El sistema global.
  - La empresa u organización global.
  - Algo activo en el mundo real que pueda participar en la tarea.
  - Un manejador artificial de todos los eventos del sistema de un caso de uso (controlador de casos de uso).

Al usar este patrón se incrementa el potencial de los elementos que pueden ser reutilizados, así como la capacidad de razonar acerca del estado actual de la actividad y la operación dentro del caso de uso actual (28).

El patrón Controlador se manifiesta en la clase Manager, pues al poseer un objeto de la clase CHC y otro de la clase PBIL, tiene la responsabilidad de controlar la ejecución de estos algoritmos.

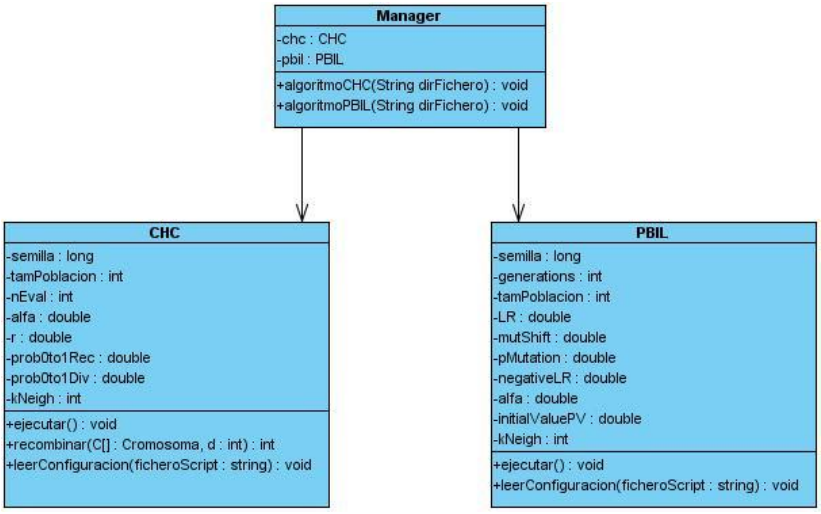


Figura 3.8: Patrón Controlador

- **Polimorfismo**

Asignar la clase (interfaz) al comportamiento y utilizar polimorfismo para implementar los comportamientos alternativos, cuando identificamos variaciones en un comportamiento. El polimorfismo tiene varios significados relacionados, entre ellos, asignar el mismo nombre a servicios en diferentes

objetos. Con la utilización de este patrón se añaden fácilmente las extensiones necesarias para nuevas variaciones y las nuevas implementaciones se pueden introducir sin afectar a los clientes (29).

Este patrón se refleja fundamentalmente en la clase que deriva de Método, puesto que tiene funcionalidades independientes y soporta el método:

```
public void leerConfiguracion(String ficheroScript)
```

Figura 3.9: Segmento de código que evidencia el patrón Polimorfismo

A continuación se expone la figura que evidencia el Patrón Polimorfismo en las clases Metodo y CHC.

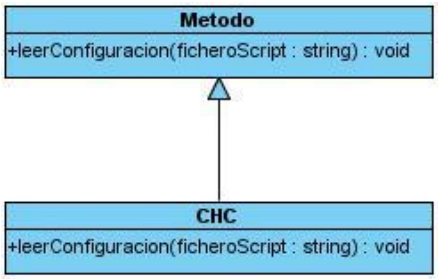


Figura 3.10: Patrón Polimorfismo

3.3 Plug-in para la Reducción de Instancias

Un *plug-in* es un módulo que añade una característica o servicio a un sistema ya existente, es decir, la aplicación principal ejecuta el nuevo componente. Los *plug-ins* le brindan a una aplicación la capacidad de agregar funcionalidades nuevas en tiempo de ejecución (25).

Como resultado de la investigación se logró implementar el *plug-in Instance Selection*, realizado con el objetivo de reducir las instancias en las muestras de la plataforma alasGRATO. El mismo es una aplicación visual que cuenta con tres pestañas, *Split Samples*, donde se le da la opción al usuario de seleccionar el fichero a reducir, además del lugar donde desea guardar las muestras divididas en un fichero para entrenamiento y otro de prueba, con los que posteriormente se realizará la reducción. En las pestañas *PBIL Algorithm* y *CHC Algorithm* se aplican los algoritmos PBIL y CHC respectivamente a las muestras divididas. En estas pestañas el usuario introduce los valores de los parámetros necesarios para la ejecución del algoritmo. Debe además seleccionar las muestras de entrenamiento y prueba ya divididas,

así como el directorio donde desea salvar los ficheros que contienen las muestras reducidas. Los anexos 3, 4 y 5 muestran cada una de las pestañas que conforman el *plug-in*.

3.4 Análisis de resultados

Los AE y de hecho, cualquier método heurístico, se caracterizan por tener un conjunto de parámetros (9, pág. 71) con valores específicos que determinan su evolución. La elección de valores apropiados para tales parámetros constituye una dificultad para la aplicación de tales algoritmos, pues comúnmente solo pueden ser seleccionados en la práctica, es decir, mediante ensayo y error, tomados de otros campos o ajustados a mano.

Basándose en lo anteriormente expuesto, los valores de los parámetros de las pruebas realizadas a los algoritmos propuestos, PBIL y CHC, fueron tomados de los experimentos del Keel correspondientes a estos algoritmos. En la siguiente tabla se muestran los valores.

PBIL		CHC	
Seed	275948593	Seed	484390552
Number of Generations	10000	Population Size	50
Population Size	50	Number of Evaluations	10000
Learning Ratio	0.25	Alfa Equilibrate Factor	0.5
Mutation Shift	0.05	Percentage of Change in Restart	0.40
Mutation Probability	0.100	Probability in Restart	0.30
Negative Learning Ratio	0.075	Probability in Diverge	0.05
Alpha Equilibrate Factor	0.5	Number of Neighbours	1
Initial Value of Probability Vector	0.3		
Number of Neighbours	1		

Tabla No. 3.1: Valores de los parámetros para los algoritmos PBIL y CHC

Luego de realizadas las pruebas con estos valores, en la siguiente tabla se reflejan los resultados obtenidos una vez aplicados los algoritmos PBIL y CHC.

Capítulo 3: Resultados y Discusión

La muestra de Cefalosporina cuenta inicialmente con 104 instancias, divididas en 83 para la muestra de entrenamiento y 21 para la muestra de prueba. Aplicando los algoritmos PBIL y CHC se obtuvo una cantidad final de 4 y 5 instancias respectivamente. De un total inicial de 315 instancias, el Ensayo_599 se divide en 252 instancias recogidas en la muestra de entrenamiento y 63 en la de prueba. Como resultado, se obtuvo una cantidad final de 11 y 9 instancias, para los algoritmos PBIL y CHC respectivamente.

Muestras	Algoritmos	Cantidad Inicial de Instancias			Cantidad Final de Instancias
		Muestra Original	Muestra de Entrenamiento	Muestra de Prueba	
Cefalosporina	PBIL	104	83	21	4
	CHC				5
Ensayo_599	PBIL	315	252	63	11
	CHC				9

Tabla No. 3.2: Pruebas a las muestras originales

Con el uso de los mismos valores tomados del Keel, manteniendo fijo algunos y variando otros que fueron ajustados a mano en dependencia del algoritmo empleado, se realizó el diseño experimental a las muestras.

3.4.1 Diseño experimental

En la tabla No. 3.3 se encuentran los valores seleccionados para hacerle pruebas al PBIL variando los siguientes parámetros.

Parámetros	Valores		
Population Size (PS)	10	30	50
Learning Ratio (LR)	0.05	0.15	0.25
Mutation Probability (MP)	0.010	0.050	0.100

Tabla No. 3.3: Niveles de los factores para el algoritmo PBIL

Capítulo 3: Resultados y Discusión

Seguidamente se exponen los resultados obtenidos luego de realizarle pruebas a las muestras originales de Cefalosporina y Ensayo_599, variando los parámetros PS entre 30 y 10, LR entre 0.15 y 0.05 y MP entre 0.050 y 0.010. Se obtuvo como resultado una cantidad final entre 5 y 24 instancias. Los mejores resultados para ambas muestras se alcanzaron al variar PS.

Muestra	Cantidad Inicial Instancias			Parámetros			Cantidad Final Instancias
	Muestra Original	Muestra Entrenamiento	Muestra Prueba	PS	LR	MP	
Cefalosporina	104	83	21	30	0.25	0.100	6
				10	0.25	0.100	5
				50	0.15	0.100	7
				50	0.05	0.100	9
				50	0.25	0.050	7
				50	0.25	0.010	7
Ensayo_599	315	252	63	30	0.25	0.100	10
				10	0.25	0.100	13
				50	0.15	0.100	15
				50	0.05	0.100	16
				50	0.25	0.050	15
				50	0.25	0.010	24

Tabla No. 3.4: Pruebas al algoritmo PBIL

Después de obtener los resultados de las pruebas aplicadas al algoritmo PBIL, en la tabla No. 5 se presentan los valores seleccionados para hacerle pruebas al CHC variando los siguientes parámetros.

Parámetros	Valores		
Population Size (PS)	10	30	50
Percentage of Change in Restart (PCR)	0.20	0.30	0.40
Probability in Restart (PR)	0.10	0.20	0.30

Tabla No. 3.5: Niveles de los factores para el algoritmo CHC

Capítulo 3: Resultados y Discusión

A continuación se presentan los resultados obtenidos luego de realizarle pruebas a las muestras originales de Cefalosporina y Ensayo_599. Al variar los parámetros PS entre 30 y 10, PCR entre 0.30 y 0.20 y PR entre 0.20 y 0.10, se obtuvo como resultado una cantidad final entre 5 y 11 instancias. Los mejores resultados se alcanzaron al variar PCR para la muestra de Cefalosporina y el PR para la muestra Ensayo_599.

Muestra	Cantidad Inicial Instancias			Parámetros			Cantidad Final Instancias
	Muestra Original	Muestra Entrenamiento	Muestra Prueba	PS	PCR	PR	
Cefalosporina	104	83	21	30	0.40	0.30	6
				10	0.40	0.30	6
				50	0.30	0.30	5
				50	0.20	0.30	5
				50	0.40	0.20	6
				50	0.40	0.10	6
Ensayo_599	315	252	63	30	0.40	0.30	10
				10	0.40	0.30	11
				50	0.30	0.30	9
				50	0.20	0.30	9
				50	0.40	0.20	8
				50	0.40	0.10	7

Tabla No. 3.6: Pruebas al algoritmo CHC

Teniendo en cuenta los resultados obtenidos en el diseño experimental para los algoritmos PBIL y CHC, se puede decir, que entre ambos no existe una diferencia significativa en cuanto al porciento de reducción de instancias, pues arrojaron como resultado, un porciento aproximado de 96 y 94 respectivamente.

Con el propósito de fijar las condiciones para los experimentos con las muestras reducidas y en busca de mejores modelos de clasificación, se realizaron pruebas de clasificación a las muestras.

3.4.2 Pruebas utilizando el clasificador Máquina de Soporte Vectorial (MSV)

Para realizar las pruebas de clasificación se emplea la Máquina de Soporte Vectorial (MSV) C-SVC perteneciente a la librería libSVM en su versión 2.8. El método empleado para determinar las muestras de entrenamiento y prueba fue la validación cruzada (*cross validation*) con valor 5, además de los parámetros *cost* con 1.0, nu con 0.5, *gamma* con 0.0 y el tipo de *kernel* utilizado fue *radial basis function: exp(-gamma* |u-v|^2)*.

En la siguiente tabla se muestran los porcientos de clasificación para las muestras completas con el objetivo de establecer una comparación luego de reducidas las instancias. También se muestra el Área ROC (30), que no es más, que la representación de la razón de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo). Los porcientos de clasificación obtenidos son de 62 y 59 para las muestras de Cefalosporina y Ensayo_599, estando en correspondencia con el Área ROC.

Muestra	Total de Instancias	Área ROC	% de Clasificación
Cefalosporina	104	0.625	62
Ensayo_599	315	0.5	59

Tabla No. 3.7: Calidad de la clasificación para las muestras completa

La tabla No. 3.8 revela los porcientos de clasificación para las mismas muestras una vez reducidas las instancias con los algoritmos PBIL y CHC sin variar los parámetros. Los porcientos de clasificación obtenidos oscilan entre 25 y 55, teniendo el mejor resultado el algoritmo CHC. Sin embargo, la muestra de Cefalosporina no está bien clasificada, pues no corresponde el Área ROC con el porciento de clasificación.

Muestra	Algoritmo	Total de Instancias	Área ROC	% de Clasificación
Cefalosporina	PBIL	4	0.25	25
	CHC	5	0.333	40
Ensayo_599	PBIL	11	0.5	54
	CHC	9	0.5	55

Tabla No. 3.8: Calidad de la clasificación para las muestras reducidas

Capítulo 3: Resultados y Discusión

Seguidamente se observan los porcientos de clasificación para las muestras Cefalosporina y Ensayo_599 respectivamente, al aplicarle el algoritmo PBIL para cada uno de los niveles de factores. Los valores obtenidos oscilan entre 40% y 71% para la muestra de Cefalosporina, estando éste último mal clasificado. El ensayo_599 arrojó resultados entre 50% y 66% de clasificación, no obstante, pese a ser éste último el más alto, el mejor clasificado tuvo un 53%.

Muestra	Parámetros			Total de Instancias	Área ROC	% de Clasificación
	PS	LR	MP			
Cefalosporina	30	0.25	0.100	6	0.667	67
	10	0.25	0.100	5	0.333	40
	50	0.15	0.100	7	0.5	71
	50	0.05	0.100	9	0.625	66
	50	0.25	0.050	7	0.5	71
	50	0.25	0.010	7	0.417	42
Ensayo_599	30	0.25	0.100	10	0.5	60
	10	0.25	0.100	13	0.5	61
	50	0.15	0.100	15	0.5	60
	50	0.05	0.100	16	0.5	50
	50	0.25	0.050	15	0.5	53
	50	0.25	0.010	24	0.5	66

Tabla No. 3.9: Calidad de la clasificación luego de ser aplicado el algoritmo PBIL

Los porcientos de clasificación arrojados luego de ejecutar el algoritmo CHC utilizando las muestras Cefalosporina y Ensayo_599, quedan reflejados en la tabla 10.

La muestra de Cefalosporina obtuvo valores entre 33% y 66% de clasificación, siendo éste primer valor, el mejor clasificado. El Ensayo_599 alcanzó valores entre 40% y 55%, éste último, el mejor clasificado.

Muestra	Parámetros			Total de Instancias	Área ROC	% de Clasificación
	PS	PCR	PR			
Cefalosporina	30	0.40	0.30	6	0.5	66
	10	0.40	0.30	6	0.333	33
	50	0.30	0.30	5	0.333	40
	50	0.20	0.30	5	0.333	40
	50	0.40	0.20	6	0.333	33
	50	0.40	0.10	6	0.333	33
Ensayo_599	30	0.40	0.30	10	0.4	40
	10	0.40	0.30	11	0.5	54
	50	0.30	0.30	9	0.5	55
	50	0.20	0.30	9	0.5	55
	50	0.40	0.20	8	0.5	50
	50	0.40	0.10	7	0.417	43

Tabla No. 3.10: Calidad de la clasificación luego de ser aplicado el algoritmo CHC

Teniendo en cuenta los resultados de clasificación luego de aplicados los algoritmos de reducción de instancias, los rangos obtenidos para PBIL oscilan entre 40% y 71%, representando este último el mayor porciento de clasificación, pese a estar mal clasificado. En el caso del algoritmo CHC los rangos están entre 33% y, 66% como mayor porciento de clasificación obtenido. Sin embargo, los mejores clasificados obtuvieron valores de 33% y 55% para las muestras de Cefalosporina y Ensayo_599 respectivamente. Debido a que las MSV trabajan mejor con muestras que tienen mayor cantidad de datos, los resultados obtenidos no fueron los mejores.

3.4.3 Pruebas no paramétricas

Las pruebas no paramétricas (31) son aquellas que no presuponen una distribución de probabilidad para los datos, son conocidas también como pruebas de distribución libre (*distribution free*). Se realizaron pruebas no paramétricas con los resultados de las muestras clasificadas aplicando el test de Wilcoxon, ya que por lo general son fáciles de usar y entender, se pueden usar con muestras pequeñas y

datos cualitativos, además de eliminar la necesidad de suposiciones restrictivas de las pruebas paramétricas (32).

Comparación	Rango Promedio	Valor	Estadística Final
PBIL vs CHC	Negativo	4.10	0.271
	Positivo	3.75	

Tabla No. 3.11: Prueba no paramétrica a la muestra Ensayo_599

Comparación	Rango Promedio	Valor	Estadística Final
PBIL vs CHC	Negativo	4.75	0.397
	Positivo	3.00	

Tabla No. 3.12: Prueba no paramétrica a la muestra Cefalosporina

Las estadísticas finales revelan que no existen diferencias significativas entre ambos algoritmos pues ninguna es menor que 0.05. Se puede determinar que PBIL tiene un valor del rango promedio equivalente a 4.10 para la muestra Ensayo_599 y de 4.75 para la muestra de Cefalosporina, siendo mayor, en ambos casos que el rango promedio de CHC.

3.5 Conclusiones Parciales

Tras el análisis de los resultados obtenidos en las pruebas, se arriba a las siguientes conclusiones:

- El algoritmo CHC presenta mayor porciento de reducción de instancias, aproximadamente de un 96%.
- La mejor calidad de clasificación la presenta el algoritmo PBIL con un 56%.
- Las diferencias entre ambos algoritmos nos son significativas.
- Los resultados obtenidos en las pruebas de clasificación no son las mejores, debido a que el clasificador Máquina de Soporte Vectorial, alcanza mejores resultados con grandes volúmenes de datos y las muestras Ensayo_599 y Cefalosporina, son relativamente pequeñas en comparación con las que soporta este clasificador. Además, puede estar dado porque los parámetros de clasificación no fueron variados.

CONCLUSIONES GENERALES

Luego de realizada la investigación se arribó a las siguientes conclusiones:

- Se realizó una búsqueda bibliográfica referente al tema de la SI, identificándose los algoritmos utilizados en el mundo para solucionar este tipo de problema, escogiéndose PBIL y CHC pertenecientes a la SP aplicada a selección de conjuntos de entrenamiento.
- Se logró implementar el *plug-in* propuesto, *Instance Selection,* para evaluar los algoritmos seleccionados, PBIL y CHC.
- Se evaluaron los algoritmos implementados con los cuales se obtuvieron resultados satisfactorios, pues al reducirse las muestras utilizando el algoritmo genético PBIL se obtuvo una reducción aproximada del 94% y para el algoritmo CHC del 96%. El mayor porciento de clasificación lo presenta el algoritmo PBIL con un 71%. Según los resultados alcanzados por ambos algoritmos, la diferencia entre estos no es significativa.

RECOMENDACIONES

Como complemento a los resultados alcanzados en esta investigación, se recomienda continuar trabajando en:

- Incluir la aplicación implementada como un servicio en la plataforma alasGRATO.
- Realizar pruebas de clasificación utilizando otro clasificador.
- Buscar nuevos métodos que contribuyan al mejoramiento de los resultados obtenidos con los algoritmos implementados.

REFERENCIAS BIBLIOGRÁFICAS

1. **Miguel Ángel Carreiro Alonso.** Farmacología. *Historia de la Farmacología.* [En línea] 08 de Enero de 2008. [Citado el: 09 de Octubre de 2009.] http://farmacologia-enfermeria.blogspot.com/2008/01/historia-de-la-frmacologa.html.
2. **Dra. Mayra Levy Rodríguez.** Farmacología. Su historia y desarrollo. [aut. libro] Colectivo de autores. *Farmacología General.* s.l. : Ciencias Médicas, 2004.
3. Servicio de Información y Noticias Científicas. *Servicio de Información y Noticias Científicas.* [En línea] 07 de Febrero de 2008. [Citado el: 09 de Octubre de 2009.] http://www.plataformasinc.es/index.php/esl/Noticias/Un-sistema-reduce-el-tiempo-de-produccion-de-farmacos.
4. Centro para la Promoción del Comercio Exterior en Cuba. *Centro para la Promoción del Comercio Exterior en Cuba*. [En línea] 2006. [Citado el: 25 de Octubre de 2009.] http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.cepec.cu%2Fcarpeta%2Foferta.doc&rct=j&q=El+grueso+de+las+exportaciones+cubanas+de+medicamentos+gen%C3%A9ricos+&ei=EoCiS-KhN4KdlgfqtHaCA&usg=AFQjCNFGiEG-NFSW_BzfLP.
5. **Tonysé de la Rosa Martín, Hermes Lázaro Herrera Martínez.** *Propuesta de algoritmos para la reducción del espacio muestral.* La Habana : s.n., 2008.
6. **José Ramón Cano de Amo.** *Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos.* Granada : Editorial de la Universidad de Granada, 2004. TIC2002-04036-C05-01.
7. **Rocío García-Durán, Fernando Fernández, Daniel Borrajo.** SCALAB. *SCALAB.* [En línea] 30 de Septiembre de 2008. [Citado el: 20 de Enero de 2010.] http://scalab.uc3m.es/~rgduran/publications/caepia07.pdf.
8. **José Ramón Cano, Francisco Herrera, Manuel Lozano.** Selección Evolutiva Estratificada de Conjuntos de Entrenamiento para la Obtención de Bases de Reglas con un Alto Equilibrio entre Precisión e Interpretabilidad. [aut. libro] José C. Riquelme y Jesús S. Aguilar-Ruiz Raúl Giráldez. *Tendencias de la Minería de Datos en España.* Granada : s.n., 2004.

Referencias Bibliográficas

9. **Enrique Yeguas Bolívar.** *Un modelo de rendimiento de algoritmos evolutivos aplicados a la selección de la solución deseada.* Granada : Editorial de la Universidad de Granada, 2009. ISBN: 978-84-692-2249-2.
10. **Pervys Rengifo, Leonardo Jiménez.** Scribd. *Scribd.* [En línea] 12 de Diciembre de 2009. [Citado el: 20 de Enero de 2010.] http://www.scribd.com/doc/12234392/Programacion-Genetica.
11. **Pedro G. Espejo, Cristóbal Romero, César Hervás, Sebastián Ventura.** KEEL. *KEEL.* [En línea] 2005. [Citado el: 25 de Enero de 2010.] http://sci2s.ugr.es/keel/pdf/keel/congreso/MAEB-Romero05.pdf. 653-660.
12. **Mauro B. G.** *Algoritmos Genéticos.* Pereira : s.n., 2010.
13. **Santana Quintero, Luis Vicente, Coello Coello, Carlos A.** Universidad Pablo D Olavide. *Universidad Pablo D Olavide.* [En línea] 02 de Diciembre de 2006. [Citado el: 03 de Febrero de 2010.] http://www.upo.es/RevMetCuant/art4.pdf ISSN: 1886-516X
14. **Yaikiel Hernández Díaz.** *Desarrollo de modelos de clasificación de actividad biológica empleando máquinas de soporte vectorial.* La Habana : s.n., 2010.
15. **Belén Melián Batista, José A. Moreno Pérez, J. Marcos Moreno Vega.** *Algoritmos Genéticos. Una visión práctica.* Canaria : s.n., 2009, Vol. 71. ISSN: 1887-1984.
16. Facultad de Ingeniería. *Facultad de Ingeniería.* [En línea] 2009. [Citado el: 8 de Febrero de 2010.] http://www.fing.edu.uy/inco/cursos/geneticos/ae/2009/Clases/clase5.pdf.
17. Intelligent Systems Group. *Intelligent Systems Group*. [En línea] 04 de Febrero de 2008. [Citado el: 13 de Febrero de 2010.] http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf
18. Computación Aplicada al Desarrollo. *Computación Aplicada al Desarrollo*. [En línea] [Citado el: 20 de Febrero de 2010.] http://www.cad.com.mx/historia_del_lenguaje_java.htm.
19. **Colectivo de Autores.** Universidad de Valencia. Universidad de Valencia. [En línea] 2004. [Citado el: 20 de Febrero de 2010.] http://www.uv.es/~jgutierr/MySQL_Java/TutorialEclipse.pdf.
20. Visual Paradigm. [En línea] Visual Paradigm Company. [Citado el: 20 de Febrero de 2010.] http://www.visual-paradigm.com/product/vpuml/.
21. CBASQA. *CBASQA*. [En línea] 02 de Septiembre de 2008. [Citado el: 20 de Febrero de 2010.] http://cbasqa.wordpress.com/2008/09/02/proceso-de-desarrollo-openup/.
22. EPF Wiki. *EPF Wiki*. [En línea] 2009. [Citado el: 20 de Febrero de 2010.] http://epf.eclipse.org/wikis/openupsp/.

Referencias Bibliográficas

23. **Guerrero, Luis A**. CC40B - Análisis y Diseño Orientado a Objetos. [En línea] [Citado el: 09 de Abil de 2010.] http://www.dcc.uchile.cl/~luguerre/cc40b/clase4.html.
24. **A. De Miguel, P. Martínez.** LaBDA. [En línea] 04 de Octubre de 2006. [Citado el: 09 de Abril de 2010.] http://basesdatos.uc3m.es/fileadmin/Docencia/DBD/Curso0607/Teoria/MODELO_ER.pdf.
25. **J. K. Pedersen.** [En línea] [Citado el: 23 de Febrero de 2010.] http://developer.kde.org/documentation/tutorials/developing-a-plugin-structure/index_es.html.
26. **Colectivo de Autores.** Entorno Virtual de Aprendizaje. [En línea] [Citado el: 15 de Abril de 2010.] http://eva.uci.cu/file.php/259/Curso_2009-2010/Conferencia_2/Conferencia_2_de_Arquitectura_2010.doc.
27. **Carlos Loayza Melicia, Loayza Villarroel Luz, Mamani Santos Nadir y Pocoata Condori Maura.** Web Docente. [En línea] [Citado el: 15 de Abril de 2010.] http://virtual.usalesiana.edu.bo/web/practica/archiv/patrones.ppt.
28. **Marcello Visconti, Hernán Astudillo.** Departamento de Informatica. Universidad Técnica Federico Santa María. [En línea] [Citado el: 15 de Abril de 2010.] http://www.inf.utfsm.cl/~visconti/ili236/Documentos/01-IntroISw.pdf.
29. **Diana Paola Hurtado Bustamante, Diana Patricia Gutiérrez Valencia, Juan Pablo Suárez Valencia, Gabriel Asakawa, Eudo Quevedo Pantoja.** Universidad del Valle. [En línea] [Citado el: 15 de Abril de 2010.] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/exposiciones2005B/polimorfismoFabricacionPura_G01/presentacion.ppt.
30. **Galparsoro I. López de Ullibarri, S. Píta Fernández.** FisTerra. [En línea] 25 de Septiembre de 2001. [Citado el: 20 de Abril de 2010.] http://www.fisterra.com/mbe/investiga/curvas_roc/curvas_roc.htm.
31. SEH-LELHA. [En línea] Sociedad Española de Hipertensión - Liga Española para la lucha contra la hipertensión arterial, 1995. [Citado el: 14 de Mayo de 2010.] http://www.seh-lelha.org/noparame.
32. El Rincón del Vago. [En línea] [Citado el: 14 de Mayo de 2010.] http://pdf.rincondelvago.com/estadistica-no-parametrica.html.

BIBLIOGRAFÍAS

- **Belén Melián Batista, José A. Moreno Pérez, J. Marcos Moreno Vega.** *Algoritmos Genéticos. Una visión práctica.* Canaria : s.n., 2009, Vol. 71. ISSN: 1887-1984.
- **Caridad Griñó, Rosa Díez, David Piñero y Jose María Ruiz Moreno.** Colegio Nacional de Ópticos Optometristas de España. [En línea] 2008. [Citado el: 14 de Mayo de 2010.] http://www.cnoo.es/modulos/gaceta/actual/gaceta445/cientifico3.pdf.
- **Carlos Loayza Melicia, Loayza Villarroel Luz, Mamani Santos Nadir y Pocoata Condori Maura.** Web Docente. [En línea] [Citado el: 15 de Abril de 2010.] http://virtual.usalesiana.edu.bo/web/practica/archiv/patrones.ppt.
- CBASQA. *CBASQA*. [En línea] 02 de Septiembre de 2008. http://cbasqa.wordpress.com/2008/09/02/proceso-de-desarrollo-openup/.
- Centro para la Promoción del Comercio Exterior en Cuba. *Centro para la Promoción del Comercio Exterior en Cuba*. [En línea] 2006. http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.cepec.cu%2Fcarpeta%2Foferta.doc&rct=j&q=El+grueso+de+las+exportaciones+cubanas+de+medicamentos+gen%C3%A9ricos+&ei=EoCiS-KhN4KdlgfqtHaCA&usg=AFQjCNFGiEG-NFSW_BzfLP.
- **Colectivo de Autores.** Entrorno Virtual de Aprendizaje. [En línea] [Citado el: 15 de Abril de 2010.] http://eva.uci.cu/file.php/259/Curso_2009-2010/Conferencia_2/Conferencia_2_de_Arquitectura_2010.doc.
- **Colectivo de Autores.** Universidad de Valencia. Universidad de Valencia. [En línea] 2004. http://www.uv.es/~jgutierr/MySQL_Java/TutorialEclipse.pdf.
- **Colectivo de Autores.** Manual del usuario de SPSS Base 14.0. Estados Unidos de América: s.n., 2005. ISBN 1-56827-683-4
- Computación Aplicada al Desarrollo. *Computación Aplicada al Desarrollo*. [En línea] http://www.cad.com.mx/historia_del_lenguaje_java.htm.
- **Diana Paola Hurtado Bustamante, Diana Patricia Gutiérrez Valencia, Juan Pablo Suárez Valencia, Gabriel Asakawa, Eudo Quevedo Pantoja.** Universidad del Valle. [En línea] [Citado el: 15 de Abril de 2010.]

Bibliografías

- http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/exposiciones2005B/polimorfismo FabricacionPura_G01/presentacion.ppt.
- **Dra. Mayra Levy Rodríguez.** Farmacología. Su historia y desarrollo. [aut. libro] Colectivo de autores. *Farmacología General.* s.l. : Ciencias Médicas, 2004.
- El Rincón del Vago**.** [En línea] [Citado el: 14 de Mayo de 2010.] http://pdf.rincondelvago.com/estadistica-no-parametrica.html.
- **Enrique Yeguas Bolívar.** *Un modelo de rendimiento de algoritmos evolutivos aplicados a la selección de la solución deseada.* Granada : Editorial de la Universidad de Granada, 2009. ISBN: 978-84-692-2249-2.
- EPF Wiki. *EPF Wiki*. [En línea] 2009. http://epf.eclipse.org/wikis/openupsp/.
- Facultad de Ingeniería. *Facultad de Ingeniería.* [En línea] 2009. http://www.fing.edu.uy/inco/cursos/geneticos/ae/2009/Clases/clase5.pdf.
- Free Download Manager. *Free Download Manager*. [En línea] 2004. http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/.
- **Galparsoro I. López de Ullibarri, S. Píta Fernández.** FisTerra. [En línea] 25 de Septiembre de 2001. [Citado el: 20 de Abril de 2010.] http://www.fisterra.com/mbe/investiga/curvas_roc/curvas_roc.htm.
- Intelligent Systems Group. *Intelligent Systems Group*. [En línea] 04 de Febrero de 2008. http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf
- **Jesús López Sánchez, Alberto Pérez de Vargas, Javier Zamora Romero, Antonio Murciano Cespedosa, Julio Alonso Fernández, Mario Reviriego Eiros y Rafael Lahoz Beltrá.** Aula Virtual de Bioestadística. [En línea] [Citado el: 20 de Abril de 2010.] http://e-stadistica.bio.ucm.es/web_spss/proc_w.html.
- **J. K. Pedersen.** [En línea] [Citado el: 23 de Febrero de 2010.] http://developer.kde.org/documentation/tutorials/developing-a-plugin-structure/index_es.html.

- **José Ramón Cano de Amo.** *Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos.* Granada : Editorial de la Universidad de Granada, 2004. TIC2002-04036-C05-01.

Bibliografías

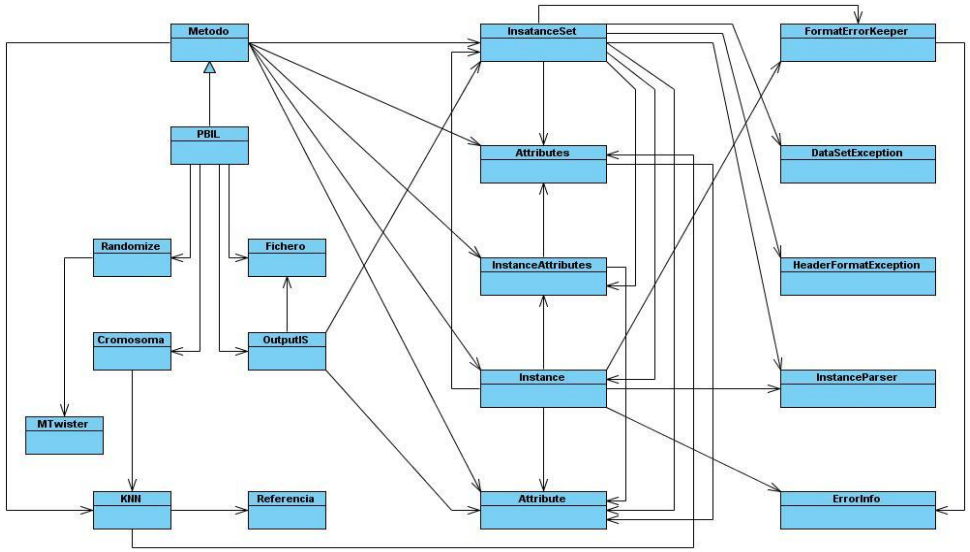
- **José Ramón Cano, Francisco Herrera, Manuel Lozano.** Selección Evolutiva Estratificada de Conjuntos de Entrenamiento para la Obtención de Bases de Reglas con un Alto Equilibrio entre Precisión e Interpretabilidad. [aut. libro] José C. Riquelme y Jesús S. Aguilar-Ruiz Raúl Giráldez. *Tendencias de la Minería de Datos en España.* Granada : s.n., 2004.
- **Magdalena Cladera Munar.** Universitat de les Illes Balears. [En línea] [Citado el: 20 de Abril de 2010.] http://www.uib.es/depart/deaweb/personal/profesores/personalpages/hdeemcm0/curs%20octubre/Apuntes%20SPSS.pdf
- **Mauro B. G.** *Algoritmos Genéticos.* Pereira : s.n., 2010.
- **Marcello Visconti, Hernán Astudillo.** Departamento de Informatica.Universidad Tecnica Federico Santa María. [En línea] [Citado el: 15 de Abril de 2010.] http://www.inf.utfsm.cl/~visconti/ili236/Documentos/01-IntroISw.pdf.
- **Miguel Ángel Carreiro Alonso.** Farmacología. *Historia de la Farmacología.* [En línea] 08 de Enero de 2008. http://farmacologia-enfermeria.blogspot.com/2008/01/historia-de-la-frmacologa.html.
- **Pedro G. Espejo, Cristóbal Romero, César Hervás, Sebastián Ventura.** KEEL. *KEEL.* [En línea] 2005. http://sci2s.ugr.es/keel/pdf/keel/congreso/MAEB-Romero05.pdf. 653-660.
- **Pervys Rengifo, Leonardo Jiménez.** Scribd. *Scribd.* [En línea] 12 de Diciembre de 2009. http://www.scribd.com/doc/12234392/Programacion-Genetica.
- **Rocío García-Durán, Fernando Fernández, Daniel Borrajo.** SCALAB. *SCALAB.* [En línea] 30 de Septiembre de 2008. http://scalab.uc3m.es/~rgduran/publications/caepia07.pdf.
- **Santana Quintero, Luis Vicente, Coello Coello, Carlos A.** Universidad Pablo D Olavide. *Universidad Pablo D Olavide.* [En línea] 02 de Diciembre de 2006. http://www.upo.es/RevMetCuant/art4.pdf ISSN: 1886-516X
- SEH-LELHA. [En línea] Sociedad Española de Hipertensión - Liga Española para la lucha contra la hipertensión arterial, 1995. [Citado el: 14 de Mayo de 2010.] http://www.seh-lelha.org/noparame.
- Servicio de Información y Noticias Científicas. *Servicio de Información y Noticias Científicas.* [En línea] 07 de Febrero de 2008. [Citado el: 09 de Octubre de 2009.] http://www.plataformasinc.es/index.php/esl/Noticias/Un-sistema-reduce-el-tiempo-de-produccion-de-farmacos.
- SPSS en Español. [En línea] 2005-2007. [Citado el: 20 de Abril de 2010.] http://www.spssfree.com/.

Bibliografías

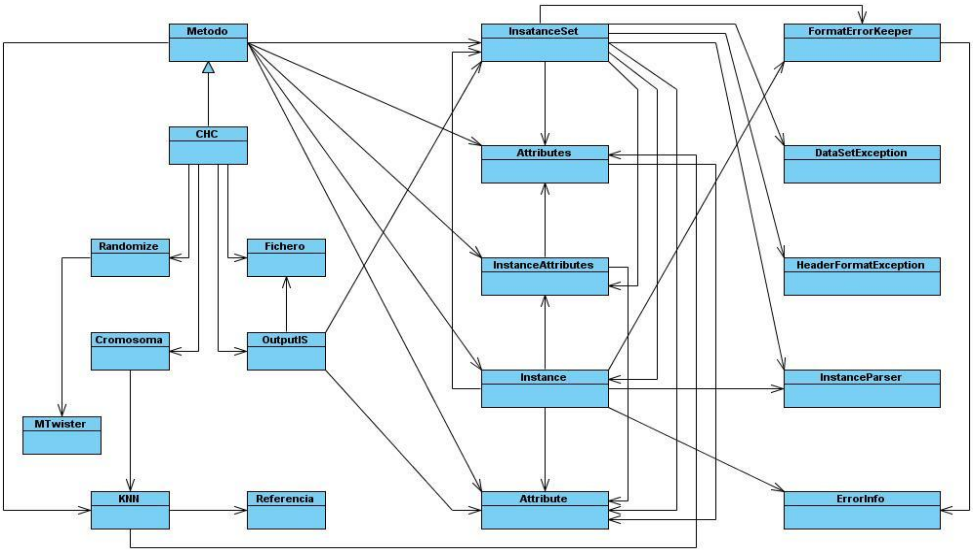
- **Tonysé de la Rosa Martín, Hermes Lázaro Herrera Martínez.** *Propuesta de algoritmos para la reducción del espacio muestral.* La Habana : s.n., 2008.
- 20. Visual Paradigm. [En línea] Visual Paradigm Company. [Citado el: 20 de Febrero de 2010.] http://www.visual-paradigm.com/product/vpuml/.
- Wilcoxon Reserch. [En línea] [Citado el: 20 de Abril de 2010.] http://www.wilcoxon.com/index.cfm.
- **Yaikiel Hernández Díaz.** *Desarrollo de modelos de clasificación de actividad biológica empleando máquinas de soporte vectorial.* La Habana : s.n., 2010

ANEXOS

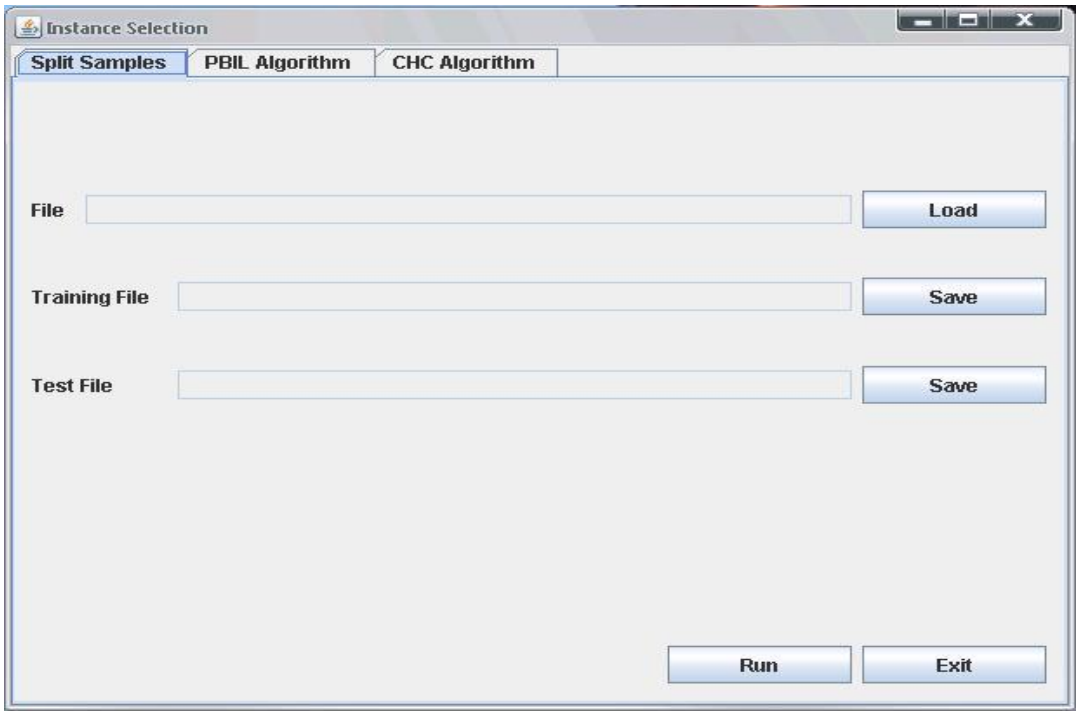
Los anexos 1 y 2 representan los diagramas de clases del diseño de los algoritmos PBIL y CHC respectivamente. Consideramos ocultar los atributos y métodos correspondientes a las clases de los diagramas, pues son muchos por cada clase y esto afectaría la comprensión de los mismos, al hacerse estos muy grandes.



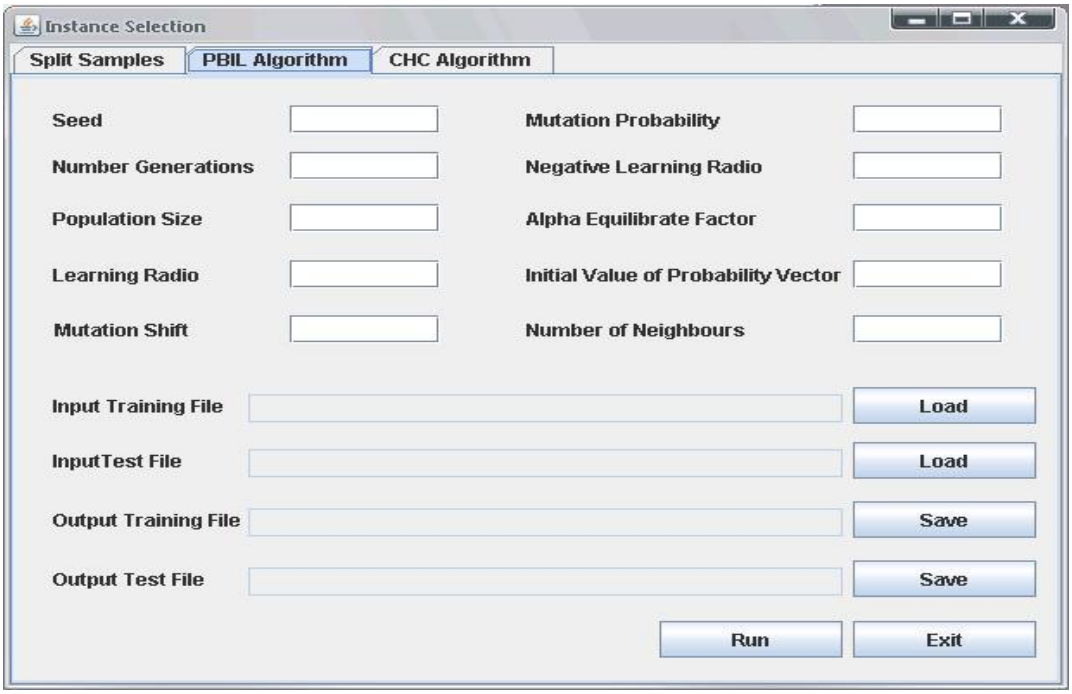
Anexo No. 1: Diagrama de clases del diseño del algoritmo PBIL



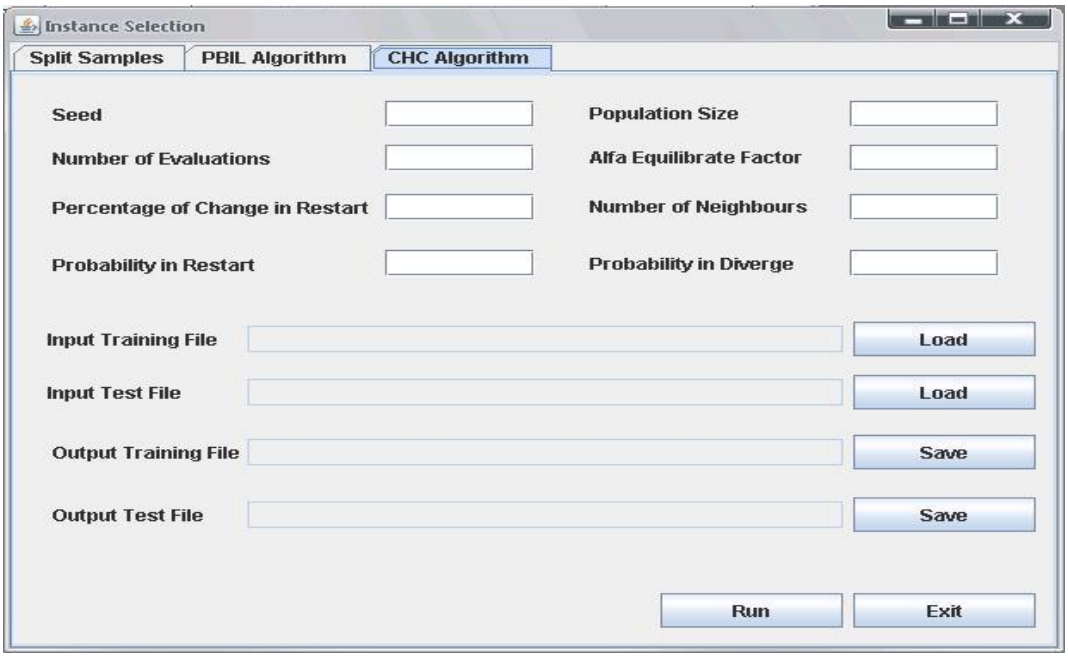
Anexo No. 2: Diagrama de clases del diseño del algoritmo CHC



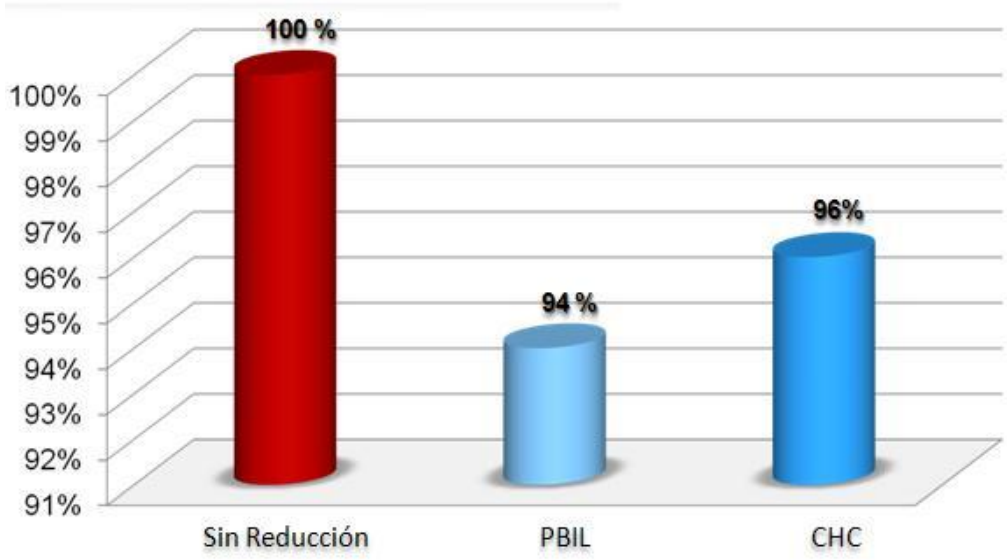
Anexo No. 3: Ventana para dividir las muestras



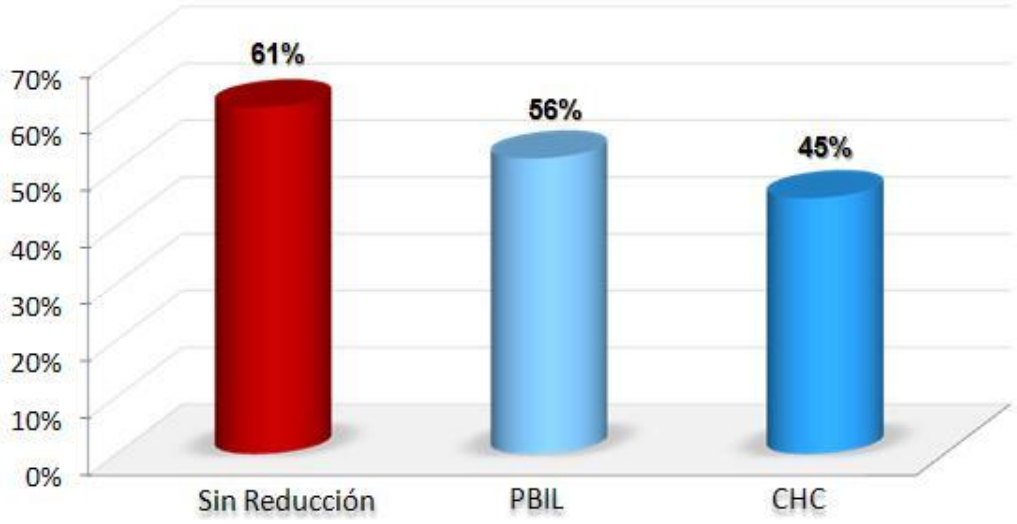
Anexo No. 4: Ventana para reducir las muestras aplicando el algoritmo PBIL



Anexo No. 5: Ventana para reducir las muestras aplicando el algoritmo CHC



Anexo No. 6: Resultados del Diseño Experimental



Anexo No. 7: Resultados del Clasificador Máquina de Soporte Vectorial

Anexo No. 8: Resultados de las Pruebas no Paramétricas

GLOSARIO DE TÉRMINOS

A

Actividad biológica: Actividad que caracteriza el comportamiento biológico en compuestos químicos (Molécula o Fragmento).

B

Bioinformática: Es la aplicación de los ordenadores y los métodos informáticos en el análisis de datos experimentales y simulación de los sistemas biológicos.

C

Compuestos Orgánicos: Compuestos cuya composición fundamental es sobre la base del elemento químico carbono.

E

Espacio muestral: Conjunto de todos los posibles resultados individuales de un experimento aleatorio.

P

Polimorfismo: Capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de diferentes tipos durante la ejecución del programa.