UNIVERSIDADE DA CORUÑA

Resolución de problemas de optimización combinatoria utilizando técnicas de computación evolutiva. Una aplicación a la biomedicina.

TESIS DOCTORAL

Vanessa Aguiar Pulido

2014

Resolución de problemas de optimización combinatoria utilizando técnicas de computación evolutiva. Una aplicación a la biomedicina

Autora: Vanessa Aguiar Pulido

Tesis Doctoral UDC / 2014

Director: Julián Dorado de la Calle

Tecnologías de la Información y las Comunicaciones

UNIVERSIDADE DA CORUÑA

Agradecimientos

Me gustaría empezar estos agradecimientos citando a Martin Luther King Jr, quién dijo una vez: "*Our lives begin to end the day we become silent about things that matter*".

Gracias a mi director de tesis, Julián, por darme la oportunidad de empezar mi carrera como investigadora. Estos siete años han permitido que, finalmente, llegase el día de hoy.

Quería agradecer también a todos aquellos que han pasado por el grupo RNASA-IMEDIR. Aunque no todos siguen aquí, algunos se han convertido en amigos que llevaré siempre en el corazón. Quería agradecer especialmente a Alejandro, su director, por ser como un padre para mí estos últimos años.

Gracias a mis compañeros del GIG, por alegrarme la vida durante estos tres últimos años. Hemos pasado muchos momentos memorables juntos y os recordaré para siempre.

Gracias a todos aquellos que han formado parte de mi "sueño americano". Ir a Stanford ha sido una experiencia que ha cambiado mi forma de ver la vida y de la que he aprendido que no se puede luchar contra lo que uno quiere. Gracias Marcos por apoyarme durante esos tres meses y a todas las personas maravillosas que he conocido y con los que he creado un vínculo.

Gracias a mis amigos de San Diego, y en especial a Connie Vera, por ser como mi segunda madre y por transmitirme su optimismo.

Finalmente, quería agradecer a mi familia, mi pareja, su familia y mis amigos, que han creído en mí aún en mis peores momentos y me han dado la fuerza necesaria para seguir adelante. Gracias por ver en mí lo que yo no soy capaz de apreciar.

A todos, muchas gracias.

« Be the change you want to see in the world »

Ghandi

Resumen

Cada día se genera una mayor cantidad de datos, tanto con respecto a su volumen como por el número de variables que involucran, lo cual representa un problema para las técnicas tradicionales. En muchos problemas el conjunto de soluciones posibles es tan elevado que la localización de una solución óptima es imposible en un tiempo razonable, por lo que es necesario emplear técnicas basadas en heurísticas. Se ha observado que las técnicas de computación evolutiva (CE) proporcionan resultados satisfactorios en situaciones en que técnicas tradicionales no los obtuvieron, en especial en su aplicación a datos biomédicos y relacionados con el diagnóstico de enfermedades.

Así, en este trabajo se ha desarrollado un modelo basado en CE capaz de, a partir de unos datos de entrada etiquetados como sujetos sanos o enfermos, extraer expresiones con las que construir un modelo de clasificación. Este modelo ha sido validado tanto contra datos sintéticos como aplicado a un conjunto de datos clínicos reales, además de comparar sus resultados con métodos similares. Es de destacar que el modelo propuesto obtiene expresiones sencillas y que logra clasificar ambos tipos de conjuntos mejor que el resto de técnicas, resultando de gran utilidad como apoyo al diagnóstico clínico.

Resumen extendido

Cada día se genera una mayor cantidad de datos, tanto con respecto a su volumen como por el número de variables que involucran. En este sentido, es necesario resaltar que el manejo de información con un número tan grande de variables representa un problema para las técnicas tradicionales. Además, en muchos problemas el conjunto de soluciones posibles es tan elevado que la localización de una solución óptima es imposible en un tiempo razonable, por lo que es necesario emplear técnicas basadas en heurísticas. Dentro de este campo, se ha observado que las técnicas de computación evolutiva (CE) proporcionan resultados satisfactorios en situaciones en que técnicas tradicionales no los obtuvieron, en especial en su aplicación a datos biomédicos y relacionados con el diagnóstico de enfermedades.

En relación con esto último, es interesante mencionar aquellos estudios que, comparando una población de sujetos sanos (controles) y enfermos (casos), traten de determinar las bases biológicas que puedan influir en el desarrollo de la enfermedad o en sufrir efectos secundarios de un fármaco concreto. Sin embargo, estas bases no siempre corresponden a una relación directa entre la genética del sujeto y el resultado observado, sino que también pueden influir otros factores en su aparición. Por ejemplo, en el caso de enfermedades complejas, además de una predisposición genética es necesario tener en cuenta lo que se conoce hoy en día como *exposoma* y que representa la variabilidad ambiental (lo que el sujeto come, inhala, el estrés sufrido, su estilo de vida…).

Así, en este trabajo se ha desarrollado un modelo basado en CE, con dos aproximaciones diferentes. Estas aproximaciones son capaces de, a partir de unos datos de entrada pertenecientes a dos grupos de pacientes (controles y casos), extraer expresiones (bien en forma de regla, bien en forma de árbol, en función de la aproximación utilizada) con las que construir un modelo de clasificación. Este modelo ha sido validado tanto contra datos sintéticos como aplicado a un conjunto de datos clínicos reales, además de comparar sus resultados con métodos similares. Como conclusión, destacar que, aplicando el método propuesto, se obtienen expresiones sencillas y modelos que son capaces de clasificar ambos tipos de conjuntos mejor que el resto de técnicas, resultando de gran utilidad como apoyo al diagnóstico clínico.

Abstract

Every day more data are being generated. Not only the volume of data increases, but also the number of variables does. This represents an issue for traditional techniques. Furthermore, many problems involve such a large set of possible solutions that finding the optimal solution in a reasonable amount of time is not feasible. Thus, using techniques based on heuristics becomes necessary. Evolutionary Computation (EC) has provided good results in situations in which traditional techniques did not, especially when applied to biomedical data and disease diagnosis.

Therefore, in this work, a model based on EC has been developed. This model, based on an input set with data that belong to healthy or diseased subjects, is capable of extracting expressions in order to build a classification model. The model proposed in this thesis has been validated on generated data, as well as applied to real clinical data, comparing the results obtained with those of other similar techniques. It is worth pointing out that the model presented extracts simple expressions and performs better when classifying both types of data sets than other existing techniques. As a result, the model presented is expected to be very useful for clinical diagnostic support.

Extended abstract

Every day more data are being generated. Not only the volume of data increases, but also the number of variables does. In this sense, it is worth highlighting that dealing with such a great number of variables represents an issue for traditional techniques. Furthermore, many problems involve such a large set of possible solutions that finding the optimal solution in a reasonable amount of time is not feasible. Thus, using techniques based on heuristics becomes necessary. Within this field, Evolutionary Computation (EC) has provided good results in situations in which traditional techniques did not, especially when applied to biomedical data and disease diagnosis.

Regarding the latter, it is interesting to mention those studies that, by comparing a population of healthy (controls) and diseased (cases) subjects, try to determine biological influence on developing a disease or on suffering side effects due to a specific drug. However, this influence does not always mean there exists a direct relationship between the genetics of an individual and an observation; other factors may intervene in its development. For example, in the case of complex diseases, in addition to genetic predisposition, what has recently been coined as *exposome* and represents environmental variability (what the subject eats, inhales, the stress suffered by the subject, his/her lifestyle…) must be taken into account.

Therefore, in this work, a model based on EC has been developed, presenting two different approaches. These approaches, based on an input set with data that belong to two different groups of subjects (controls and cases), are capable of extracting expressions (represented as rules or as trees depending on the approach) in order to build a classification model. The model proposed in this thesis has been validated on generated data, as well as applied to real clinical data, comparing the results obtained with those of other similar techniques. Finally, it is worth pointing out that the method presented extracts simple expressions and models which perform better when classifying both types of data sets than other existing techniques. As a result, the model presented is expected to be very useful for clinical diagnostic support.

Resumo

Cada día xérase unha maior cantidade de datos, tanto con respecto ao seu volume como polo número de variables que involucran, o cal representa un problema para as técnicas tradicionais. En moitos problemas o conxunto de solucións posibles é tan elevado que a localización dunha solución óptima é imposible nun tempo razoable, polo que é necesario empregar técnicas baseadas en heurísticas. Observouse que as técnicas de computación evolutiva (CE) proporcionan resultados satisfactorios en situacións en que técnicas tradicionais non os obtiveron, en especial na súa aplicación a datos biomédicos e relacionados co diagnóstico de enfermidades.

Así, neste traballo desenvolveuse un modelo baseado en CE capaz de, a partir duns datos de entrada etiquetados como suxeitos sans ou enfermos, extraer expresións coas que construír un modelo de clasificación. Este modelo foi validado tanto contra datos sintéticos como aplicado a un conxunto de datos clínicos reais, ademais de comparar os seus resultados con métodos similares. Compre destacar que o modelo proposto obtén expresións sinxelas e que logra clasificar ambos tipos de conxuntos mellor co resto de técnicas, resultando de gran utilidade como apoio ó diagnóstico clínico.

Resumo extendido

Cada día xérase unha maior cantidade de datos, tanto con respecto ao seu volume como polo número de variables que involucran. Neste sentido, é preciso resaltar que o manexo de información cun número tan grande de variables representa un problema para as técnicas tradicionais. Ademais, en moitos problemas o conxunto de solucións posibles é tan elevado que a localización dunha solución óptima é imposible nun tempo razoable, polo que é necesario empregar técnicas baseadas en heurísticas. Dentro deste campo, observouse que as técnicas de computación evolutiva (CE) proporcionan resultados satisfactorios en situacións en que técnicas tradicionais non os obtiveron, en especial na súa aplicación a datos biomédicos e relacionados co diagnóstico de enfermidades.

En relación con isto último, é interesante mencionar aqueles estudos que, comparando unha poboación de suxeitos sans (controis) e enfermos (casos), traten de determinar as bases biolóxicas que poidan influír no desenvolvemento da enfermidade ou en sufrir efectos secundarios dun fármaco concreto. Sen embargo, estas bases non sempre corresponden a unha relación directa entre a xenética do suxeito e o resultado observado, senón que tamén poden influír outros factores na súa aparición. Por exemplo, no caso das enfermidades complexas, ademais dunha predisposición xenética é necesario ter en conta o que hoxe en día se coñece como *exposoma* e que representa a variabilidade ambiental (o que o suxeito come, inhala, o estrés sufrido, o seu estilo de vida…).

Así, neste traballo desenvolveuse un modelo baseado en CE, con dúas aproximacións diferentes. Estas aproximacións son capaces de, a partir duns datos de entrada pertencentes a dous grupos de pacientes (controis e casos), extraer expresións (ben en forma de regra, ben en forma de árbore, en función da aproximación utilizada) coas que construír un modelo de clasificación. Este modelo foi validado tanto contra datos sintéticos como aplicado a un conxunto de datos clínicos reais, ademais de comparar os seus resultados con métodos similares. Como conclusión, destacar que, aplicando o método proposto, se obteñen expresións sinxelas e modelos que son capaces de clasificar ambos tipos de conxuntos mellor co resto de técnicas, resultando de gran utilidade como apoio ó diagnóstico clínico.

Prólogo

1. Motivación

Un problema de optimización combinatoria (Cook et al., 1997), generalmente, se caracteriza por un conjunto finito de soluciones admisibles, Ω, y una función objetivo, f: Ω → IR, que asocia un valor a cada solución admisible. La resolución del problema consiste en determinar la(s) solución(es) de Ω minimizando o maximizando f. Existen múltiples ejemplos de problemas de optimización combinatoria, entre los que se encuentra la extracción de conocimiento, que puede ser modelada de esta forma.

El crecimiento extremadamente rápido de los datos almacenados en las bases de datos y la necesidad de reacciones eficaces por parte de los que toman las decisiones frente a estas nuevas informaciones ha estimulado el desarrollo rápido de la extracción de conocimiento a partir de datos (en inglés *Knowledge Discovery in Databases* (Frawley et al., 1991)). Esto, no es un proceso trivial de identificación de estructuras desconocidas, válidas y potencialmente explotables, sino que está compuesto por diversas fases, cada una con su propia complejidad, enumeradas a continuación: adquisición y almacenamiento de datos (*Data Warehousing*), pre-procesado de los datos, minería de datos (*Data Mining*) y post-procesado. Este trabajo se centrará principalmente en el tercer grupo (minería de datos), el cual integra la elección de la modelización adecuada y el método que se utilizará con su aplicación a la búsqueda de estructuras de datos subyacentes y a la creación de modelos explicativos y/o predictivos.

La etapa de minería de datos (Fayyad et al., 1996) puede, a su vez, dividirse en tres bloques: la clasificación supervisada, la categorización o clustering y la búsqueda de reglas de asociación. Estas tareas se pueden modelar como problemas de optimización combinatoria.

Uno de los objetivos de la bioinformática es realizar una aplicación de lo anterior a estudios genéticos como pueden ser los *estudios de asociación*, en los que se compara el ADN de

personas sanas con el ADN de personas enfermas para tratar de encontrar relaciones entre mutaciones en dicho ADN y la enfermedad. Se suelen estudiar mutaciones presentes en al menos un 1% de la población, denominadas SNPs (Single Nucleotide Polymorphisms) (den Dunnen and Antonarakis, 2000), cuyo número supera los diez millones para los seres humanos. Por lo tanto, será necesario tratar una gran cantidad de datos obtenidos a partir del genotipado del ADN de pacientes, en los cuales deben considerarse numerosas variables.

Estos estudios, además, se pueden llevar a cabo sobre enfermedades complejas (Motulsky, 2006), en las cuales, además de influir una posible predisposición genética, afectan factores ambientales (*exposoma*), por lo que una persona aparentemente sana puede estar predispuesta genéticamente pero no desarrollar la enfermedad. Debido a la naturaleza de estas enfermedades es difícil establecer una relación gen-enfermedad, puesto que, generalmente, una enfermedad de este tipo se deberá al efecto conjunto de varios SNPs con un efecto muy bajo por separado. Estas enfermedades tienen un gran impacto sanitario y entre ellas se encuentran: el cáncer, enfermedades mentales, enfermedades cardíacas…

Para realizar este tipo de estudios existen dos aproximaciones posibles a la hora de escoger los individuos que formarán parte del estudio:

- *Aproximación por genes candidatos:* Esta aproximación se basa en hipótesis causativas o genes candidatos, es decir, hipótesis previas. Tiene la ventaja de que ofrece la posibilidad de estudiar de forma precisa las bases genéticas en torno a un trastorno en cuestión.
- *Genome Wide Association Studies (GWAs*, también *Whole Genome Analysis* o *WGA):* En este caso se consideran regiones génicas, se tiene en cuenta la densidad de SNPs entre el número de muestras y suelen estar basados en estudios de ligamiento. Tienen la ventaja, con respecto a la aproximación anterior, de que se elimina el sesgo previo de tener unos genes que, tal vez, no son los adecuados.

Debido a las limitaciones tecnológicas existentes (en cuanto a la extracción de información genética relevante a partir del ADN de cada paciente) se tendía más a la primera aproximación dado que no era posible obtener un número muy alto de variables en un mismo proceso, pero, actualmente, se tiende más hacia los GWAs (Kreiner and Tillman Buck, 2005), ya que cada vez es posible obtener mayor cantidad de información del

genoma. Sin embargo, en el caso de ciertas enfermedades, sigue interesando centrarse en una serie de genes concretos que se presuponen de interés.

Además de aplicar métodos de minería de datos a enfermedades complejas, también sería posible la búsqueda de relaciones entre SNPs y efectos secundarios importantes de medicamentos (por ejemplo, que podrían llegar a provocar incluso la muerte), lo cual, en ámbitos farmacéuticos, tiene gran importancia debido a las graves consecuencias que pueden tener (Chamberlain, 2008).

El manejar información con un número tan grande de variables representa un problema para las técnicas tradicionales. A la hora de resolver problemas de optimización combinatoria se pueden considerar dos tipos de algoritmos:

- *Exactos:* Son capaces de localizar la solución óptima, aquella en la que la función objetivo alcanza su valor extremo.
- *Heurísticos:* Encuentran una buena solución, pero no se garantiza que sea la óptima.

En muchos problemas el conjunto de soluciones posibles es tan elevado que la localización de una solución óptima es imposible en un tiempo razonable. Por ello, se opta por algoritmos heurísticos (Pearl, 1984) que encuentran una buena solución. Cada método se enfrentará al problema desde una perspectiva distinta. Entre los métodos heurísticos se encuentran las técnicas basadas en Inteligencia Artificial.

Las técnicas tradicionales de Inteligencia Artificial (IA) poseen ciertas restricciones, como pueden ser la necesidad de poseer cierta información genética con un coste prohibitivo, no ser capaz de procesar más de una cierta cantidad de datos o no ser capaz de inferir ciertos parámetros que deberán ser introducidos de forma manual. Sin embargo, se ha observado que, dentro de las técnicas de Inteligencia Artificial, las técnicas de Computación Evolutiva (CE) han proporcionado resultados satisfactorios en situaciones en que técnicas tradicionales no los obtuvieron.

La Computación Evolutiva (Holland, 1975) constituye una serie de métodos de búsqueda y optimización que imitan los principios de la evolución natural. Dentro de la Computación Evolutiva se incluyen técnicas como: Algoritmos Genéticos, Estrategias Evolutivas, Programación Evolutiva y Programación Genética (Koza, 1992).

Los algoritmos evolutivos, como se denomina de forma genérica a la implementación de los diferentes métodos de la Computación Evolutiva, proporcionan una técnica de optimización universal aplicable a problemas de diferentes dominios como: optimización de parámetros, búsqueda, problemas de combinatoria y generación automática de programas. A diferencia de métodos diseñados para tareas específicas de optimización, estos no requieren más conocimiento específico del problema que una función objetivo. Se distinguen por su robustez y proporcionan una aproximación eficiente y eficaz para manipular espacios de búsqueda grandes, complejos y pobremente definidos, donde métodos de búsqueda heurística o enumerativa son incapaces.

La elección de técnicas de Computación Evolutiva para problemas de optimización combinatoria aplicada a datos biomédicos se fundamenta, además de lo anterior, en que estas técnicas pueden ofrecer unos resultados satisfactorios cuando se aplican en el procesamiento de grandes volúmenes de datos, en especial datos médicos y relacionados con el diagnóstico de enfermedades.

Por esto, en esta tesis se propone aplicar un modelo basado en Computación Evolutiva para problemas de optimización combinatoria a datos genéticos de pacientes gallegos relacionados con la esquizofrenia, haciendo posible un tratamiento automático de estos datos. Para esto será necesario modelar el problema siguiendo la aproximación de optimización combinatoria.

2. Objetivos de la investigación

2.1. Objetivo general

En diversos estudios genéticos se observó que en los datos procedentes del ADN de numerosos pacientes existían relaciones entre algunas variables y el hecho de padecer, posteriormente, una enfermedad concreta. Esas variables, en muchos casos, eran mutaciones o, más concretamente, SNPs. Este es el caso de las enfermedades complejas. Sin embargo, como se ha mencionado anteriormente, cada SNP tiene un efecto muy bajo por separado, por lo que será necesario el efecto conjunto de varios SNPs para que exista predisposición a padecer una enfermedad compleja. Además, existen factores

medioambientales que pueden enmascarar los efectos de los SNPs, por lo que será necesario poseer más casos y representará un problema para las técnicas tradicionales.

Dado que técnicas tradicionales de minería de datos no han obtenido resultados concluyentes en este dominio y la Computación Evolutiva se ha mostrado eficaz en la resolución de problemas en que las técnicas anteriores no lo eran, se propone la utilización de este tipo de técnicas para resolver el problema tratado.

Esta tesis, por lo tanto, se centrará en el estudio de las posibilidades que ofrecen las técnicas de Computación Evolutiva para el desarrollo de un método que permita extraer estructuras subyacentes en datos de forma automatizada. Adicionalmente, se pretende generar conocimiento aplicable en biomedicina sobre la predisposición genética a padecer enfermedades complejas para tratar de ayudar a mejorar la prevención y el diagnóstico de la enfermedad estudiada o las posibles interacciones de un fármaco dependiendo del ADN de un paciente.

Este objetivo general se puede desglosar en los objetivos específicos detallados a continuación.

2.1.1. Objetivos específicos

- Estudiar en profundidad los métodos existentes para la clasificación en base a la extracción de estructuras (por ejemplo, reglas o árboles) en conjuntos con gran cantidad de datos en los que intervienen muchas variables, aplicados, especialmente, en el ámbito biomédico.
- Diseñar un nuevo método de minería de datos obtenido de la combinación, ampliación o adaptación de distintas técnicas de Computación Evolutiva a problemas de optimización combinatoria. Los datos procedentes de genotipar el ADN tienen un gran tamaño y suele ser necesario tener en cuenta numerosas variables, por lo tanto, el espacio de conocimiento potencial tendrá complejidad exponencial, formando parte este tipo de problemas de los de optimización NP-complejos y siendo, por lo tanto, más adecuada la elección de métodos heurísticos.
- Estudiar diversos tipos de representaciones para la modelización de las estructuras subyacentes en los datos. Estas estructuras representan combinaciones de variables con impactos limitados que, en conjunto, tienen un impacto más grande.

- Estudiar el comportamiento del método diseñado a la hora de clasificar en base a las estructuras que es capaz de extraer sobre diversos conjuntos de datos.
- Estudiar la verificación de las hipótesis planteadas.
- Comparar el método desarrollado con otras técnicas similares.
- El objetivo final es generar información relevante en ámbitos biomédicos, de forma que se ayude a mejorar la prevención y el diagnóstico de enfermedades complejas o las interacciones con medicamentos, por ello se realizará una aplicación a un caso real.

3. Estructura de la tesis

La presente tesis está estructurada en varios capítulos de forma que, partiendo de una serie de descripciones sobre los métodos y técnicas existentes actualmente, se analizan las diversas formas de conseguir los diferentes objetivos que se acaban de presentar.

- El presente prólogo sirve para introducir el ámbito del problema sobre el que se desarrollará el trabajo, así como para formular los objetivos que se pretenden alcanzar.
- En el capítulo I se introducen una serie de conceptos biológicos necesarios para una mejor comprensión de la presente tesis y se realiza una descripción de las técnicas de Inteligencia Artificial en las que se basa este trabajo. Se aportarán nociones sobre Algoritmos Genéticos, entrando en el campo de la Computación Evolutiva, su funcionamiento y cómo, consecuencia de una evolución de esta técnica, surgió la Programación Genética, así como el funcionamiento de la misma.
- En el capítulo II se muestran los trabajos existentes en los campos que atañen a esta tesis: desarrollo de métodos que se han aplicado a la resolución del mismo tipo de problema y, en especial, aquellos que utilizan CE.
- Partiendo del análisis y estudio de los métodos existentes relacionados con el presente campo, el capítulo III plantea la hipótesis de trabajo que se ha tomado para la realización de esta tesis.
- En el capítulo IV se describe detalladamente el modelo propuesto en esta tesis. En el capítulo V se muestran los resultados obtenidos tras aplicar las distintas

aproximaciones del modelo propuesto al campo de la biomedicina, mientras que el capítulo VI incluye los resultados de comparar dichas aproximaciones con diversos métodos de minería de datos similares.

- En el capítulo VII se muestran las conclusiones obtenidas tras la realización de la presente tesis y el capítulo VIII contiene las conclusiones en inglés, atendiendo a lo exigido en la normativa vigente sobre la mención internacional en el título de Doctor.
- En el capítulo IX se plantean posibles líneas futuras de investigación a partir del trabajo presentado.
- Finalmente, se incluye la bibliografía, un glosario de términos, un anexo con los trabajos publicados más relevantes referentes a esta tesis y un índice general de términos.

Índice de Contenidos

Índice de figuras

Índice de tablas

Lista de abreviaturas

ADN	Ácido desoxirribonucleico
AG	Algoritmo Genético. En plural: AA.GG. En inglés: *Genetic Algorithm* (GA)
CE	Computación evolutiva En inglés: *Evolutionary Computation* (EC)
FN	Falso negativo
FP	Falso positivo
IA	Inteligencia Artificial
IRL	Iterative Rule Learning
OR	Odds ratio
PG	Programación Genética En inglés: *Genetic Programming* (GP)
PV-	Valor predictivo negativo
PV+	Valor predictivo positivo

SNP	Single Nucleotide Polymorphism
VN	Verdadero negativo
VP	Verdadero positivo

I. Fundamentos

1. Fundamentos biológicos

Para comprender mejor lo tratado en esta tesis, a continuación se explicarán algunos conceptos de genética y de medicina relacionados con su contenido.

1.1. Conceptos relativos a la genética

Para poder estudiar la predisposición genética de un individuo a padecer una enfermedad con componente genético, es necesario estudiar su genoma. El genoma es el conjunto del material genético contenido en una célula. Se presenta en forma de una larga molécula, el ADN (Watson and Crick, 1953), enroscada en el núcleo de cada célula del organismo. El ADN está formado por la unión de pequeñas moléculas denominadas nucleótidos. En el ADN, sólo existen cuatro tipos de nucleótidos distintos, diferenciándose solamente en uno de sus componentes, las llamadas bases nitrogenadas. En la Figura 1, se muestra una ilustración del ADN.

En las moléculas de ADN, además, se encuentran las unidades hereditarias llamadas genes, las cuales son parte de un elemento más grande, el cromosoma. Los cromosomas (Nägeli, 1842) son estructuras en forma de bastón que aparecen en el momento de la reproducción celular, en la división del núcleo. Están constituidos químicamente por ADN más unas proteínas específicas. Su número es constante en todas las células de un individuo pero varía según las especies, en los seres humanos es 46. Un cromosoma está formado por dos cromátidas (dos hebras de ADN idénticas) que permanecen unidas por un centrómero, como se puede ver en la Figura 2.

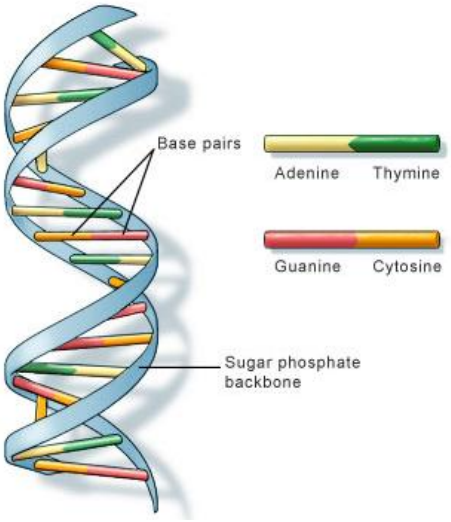


Figura 1. ADN

El ser humano tiene veintitrés pares de cromosomas homólogos. Un cromosoma homólogo es cada uno del par de cromosomas que empareja entre sí durante un tipo de división celular. Estos cromosomas suelen tener igual disposición de secuencia de ADN de un extremo a otro y, por ello, de genes, es decir, tienen información para los mismos caracteres, lo cual no significa que lleven la misma información genética.

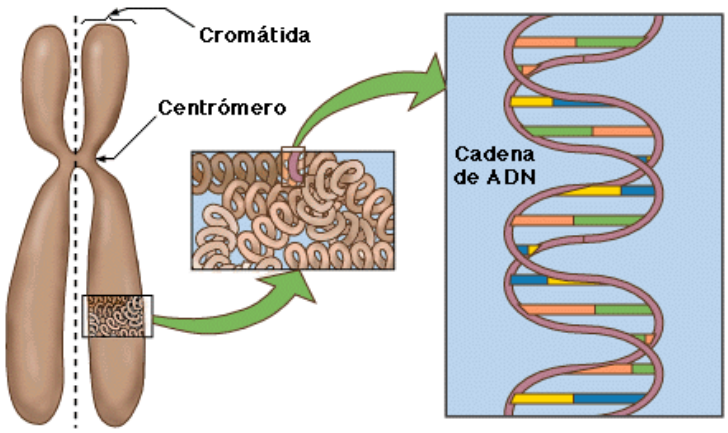


Figura 2. Cromosoma

Un gen (De Vries, 1889) es una unidad de información localizada en un cromosoma concreto y está constituido por un fragmento de ADN, esto se puede ver más claramente en la Figura 3. Cada gen dirige la producción de una o varias proteínas y asegura la transmisión de un carácter dado. El genoma humano contiene entre 20 y 25.000 genes repartidos en 24 pares de cromosomas.

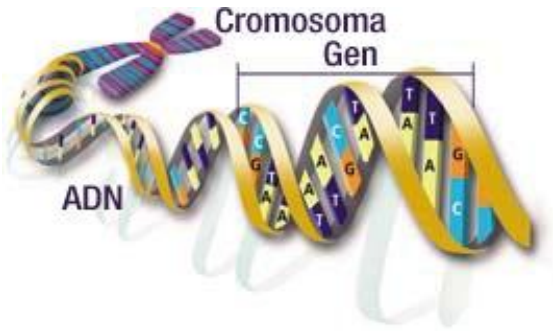


Figura 3. Cromosoma, ADN y gen

En cada individuo, cada gen es representado por dos alelos situados en el mismo locus (lugar físico que un gen ocupa en un cromosoma). Un alelo es, pues, una versión posible de un locus o cada forma diferente que puede tener un gen, constituido por un trozo de ADN. Con el objetivo de que se comprenda mejor este concepto, se muestra la Figura 4.

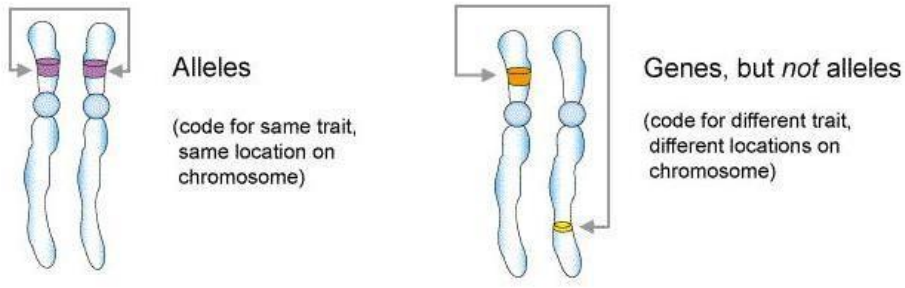


Figura 4. Alelos

Cuando los dos alelos son idénticos en cuanto a su composición, se considera que el individuo es homocigótico para ese gen y, en caso contrario, el individuo es considerado heterocigótico para ese gen.

La predisposición a padecer ciertas enfermedades no viene determinada por un gen, sino que se piensa que es debida a la interacción de varios fragmentos de genes o SNPs, como por ejemplo el caso real al que se ha aplicado el modelo propuesto en esta tesis, la esquizofrenia.

Un SNP, mostrado en la Figura 5, es un pequeño cambio genético o variación que puede ocurrir en el genoma de una persona (Britannica, 2014). El código genético se especifica con las letras que representan las iniciales de los cuatro nucleótidos: A (adenina), C (citosina), T (timina) y G (guanina). Una variación ocurre, pues, cuando un nucleótido, por

ejemplo A, reemplaza a uno de los tres restantes, C, G o T. Para que una variación sea considerada un SNP debe estar presente en, al menos, 1% de la población.

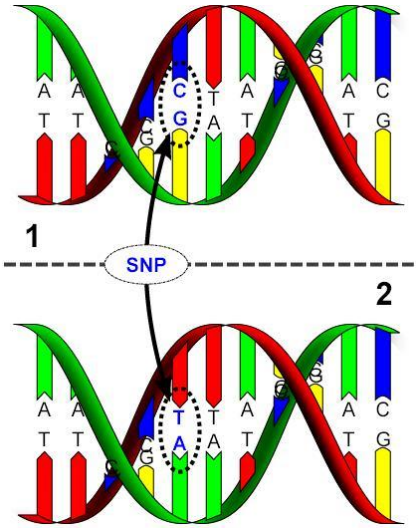


Figura 5. SNP

A pesar de que muchos SNPs no producen cambios físicos en las personas, los científicos creen que ciertos SNPs pueden influir en la predisposición a padecer una enfermedad e incluso influenciar en la respuesta que una persona pueda tener a un tratamiento farmacológico. Además, los SNPs encontrados en el genoma tienen una mayor probabilidad de alterar la función biológica de una proteína.

Cuando se tiene un conjunto de SNPs en una sola cromátida que se encuentran estadísticamente asociados, esto se denomina haplotipo. Se piensa que estas asociaciones, y la identificación de algunos alelos de un haplotipo, permiten identificar el resto de posiciones polimórficas en su región. Tal información es de mucho valor para la investigación de la genética tras enfermedades comunes, además de para su recopilación en el Proyecto Internacional HapMap (HapMap, 2014).

La definición más habitual de haplotipo, sin embargo, es "constitución genética de un cromosoma individual". Un haplotipo es, pues, una combinación de alelos ligados a múltiples locus que se transmiten juntos. El haplotipo se puede referir a un solo locus o a un genoma completo. En el caso de organismos diploides como el hombre, un haplotipo del genoma comprende sólo un miembro del par de alelos en cada locus (es decir, la mitad de un genoma diploide, se puede decir que es un sinónimo de "genotipo haploide").

1.2. Enfermedades de base genética y su estudio

A la hora de estudiar la relación entre el genoma de una persona y la predisposición a padecer una enfermedad, se considerarán dos tipos de enfermedades: las monogénicas y las complejas.

Las enfermedades monogénicas son aquellas en las que existe una relación directa entre un gen y la enfermedad. Tienen un bajo impacto sanitario, presentan mayoritariamente herencia mendeliana (forma en que se transmiten los genes y por ende los rasgos de padres a hijos (NHGRI, 2014)), poseen una alta penetrancia de alelos implicados y una baja frecuencia de variantes implicadas. Entre las enfermedades monogénicas se encuentran: la fibrosis quística, MCH familiar, la enfermedad de Huntington…

Por el contrario, las enfermedades complejas son aquellas en las que establecer una relación gen-enfermedad resulta difícil. Estas enfermedades, además, tienen un gran impacto sanitario, no presentan herencia mendeliana, poseen una baja penetrancia de alelos implicados y una alta frecuencia de variantes implicadas. Entre las enfermedades complejas se encuentran: el cáncer de mama, algunas enfermedades cardíacas, la esquizofrenia…

Para entender mejor lo anterior, es necesario explicar algunos términos utilizados:

- ✓ La *penetrancia* genética indica, en una población, la proporción de individuos que presentan un genotipo causante de enfermedad y que expresan el fenotipo patológico. Cuando esta proporción es inferior al 100%, se considera que el genotipo patológico tiene una penetrancia reducida o incompleta. La penetrancia se define para cada alelo de un gen. Por tanto, se corresponde con el porcentaje de veces que un alelo determinado de un gen produce el fenotipo con el que se le ha asociado. La penetrancia incompleta se debe a que el fenotipo asociado puede tener una causa multifactorial y por tanto, su expresión fenotípica puede verse inhibida por otros factores, ya sean ambientales o genéticos.
- ✓ El *genotipo* es el conjunto de genes que contiene un organismo heredado de sus progenitores, mientras que el *fenotipo* es la manifestación externa del genotipo, es decir, la suma de los caracteres observables en un individuo. El fenotipo es el resultado de la interacción entre el genotipo y el ambiente (Mendel, 1865).

La diferencia se puede ver en la siguiente figura:

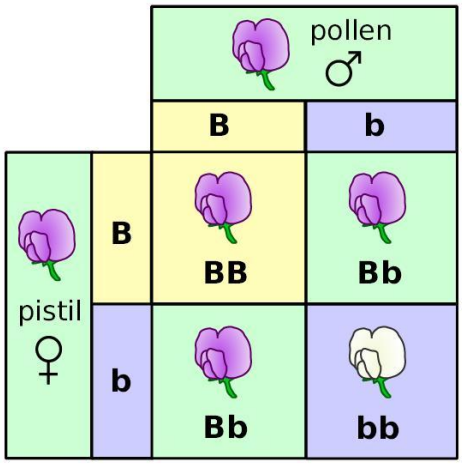


Figura 6. Genotipo y fenotipo

Para el fenotipo "violeta", se tendrían dos posibles genotipos: BB y Bb, mientras que para el fenotipo "blanco", se tendría un único genotipo posible: bb.

Debido al origen multifactorial de la esquizofrenia, el modo de herencia es complejo, y no sigue el clásico patrón de herencia mendeliana (McGue and Gottesman, 1989; McGuffin et al., 1995). El *fenotipo esquizofrenia* sería el resultado de la interacción entre factores genéticos y ambientales. Los estudios familiares, estudios de gemelos y estudios de adopción (Cardno and Gottesman, 2000; Ingraham and Kety, 2000; Kendler and Diehl, 1993) constituyen la primera prueba de la presencia de factores genéticos en el origen y la evolución de la esquizofrenia. Los estudios de ligamiento han proporcionado los datos de las regiones cromosómicas diana en la genética de la esquizofrenia y, gracias a los estudios de asociación, se han relacionado diversos SNPs con mayor riesgo de presentar esquizofrenia. A pesar del riesgo relativo asociado a cada SNP de forma aislada no es importante, se estima que el conjunto de las alteraciones genéticas supone entre el 65 y el 85% del riesgo total de presentar el trastorno (Chinchilla Moreno, 2007).

A continuación, se tratarán en mayor profundidad las enfermedades complejas dado que la presente tesis está relacionada con la esquizofrenia, la cual, como se ha mencionado anteriormente, forma parte de este tipo de enfermedades.

1.2.1. Las enfermedades complejas

En la enfermedades complejas, el fenotipo de los caracteres complejos está sometido a gran variabilidad, como pueden ser: la edad, la presión arterial, el colesterol en sangre, el índice de masa corporal… (Brión, 2008)

Además, debe tenerse en cuenta que la heredabilidad de un carácter es la proporción de variabilidad que puede ser atribuida a factores genéticos, así se obtiene que:

$$Variabilidad\ fenotípica = variabilidad\ genotípica + variabilidad\ ambiental\ (exposoma)$$

A la hora de realizar estudios para tratar de ver qué genes afectan en la predisposición a una enfermedad se seguiría el siguiente flujo de investigación:

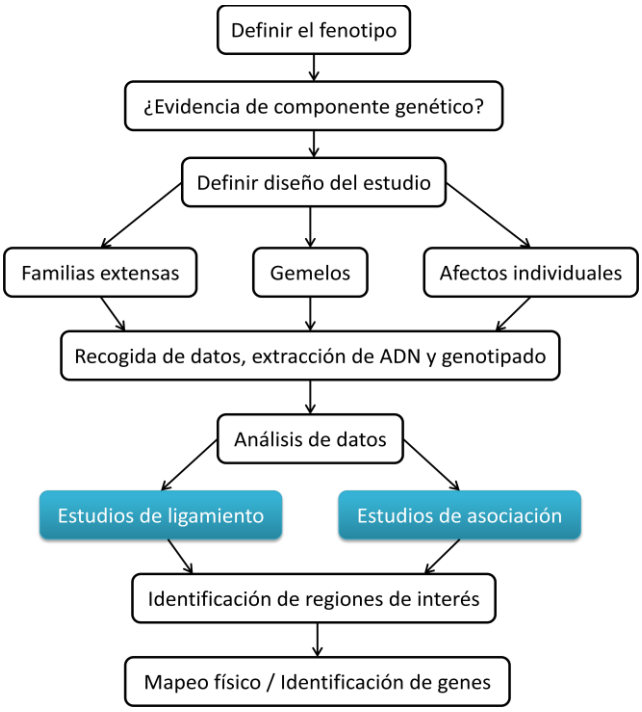


Figura 7. Flujo de investigación

Por genotipado se entiende el proceso de determinación del genotipo de un individuo mediante una prueba biológica. Entre los métodos con los que se cuenta actualmente para efectuarlo está la secuenciación de ADN. La secuenciación de ADN es un conjunto de métodos y técnicas bioquímicas cuya finalidad es la determinación del orden de los nucleótidos (A, C, G y T) en una secuencia corta de ADN, como se puede ver en la Figura 8.

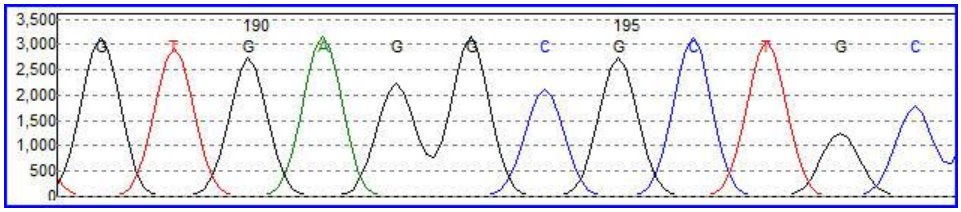


Figura 8. Secuenciación de ADN

En lo que atañe a esta tesis, se supone que ya se tienen los datos, resultado de realizar un genotipado por secuenciación de ADN, y que se va a proceder a un análisis de dichos datos. Para ello, como se puede ver en la Figura 7, existen dos posibles métodos: los estudios de ligamiento y los estudios de asociación. Dado que una de las principales limitaciones que tienen los estudios de ligamiento en el caso de las enfermedades genéticamente complejas, como la esquizofrenia, es que tienen poco poder para detectar genes de efecto menor, en este caso los datos de entrada fueron proporcionados siguiendo el diseño de un estudio de asociación.

1.2.2. Estudios de asociación

En un estudio de asociación genética se compara la frecuencia con la que se encuentran los alelos de un SNP concreto en personas afectadas por una misma enfermedad y no emparentadas, con la frecuencia observada en individuos sanos y no emparentados del mismo grupo étnico y origen geográfico, los cuales constituyen la población de control (Sáiz and Fañanás, 1998).

Al realizar un estudio de asociación en poblaciones, se tienen tres posibilidades a la hora de realizar el diseño, dependiendo de cómo se escoge la población:

- ✓ Diseño transversal: se selecciona la población sin criterios previos, o con criterios no relacionados ni con el factor de riesgo (variable asociada con un aumento de riesgo de padecer la enfermedad) ni con la enfermedad.
- ✓ Estudio de cohortes: se selecciona la población en función del factor de riesgo (genotipo).
- ✓ Estudio caso – control: se selecciona la población en función de la enfermedad.

Este último tipo suele ser el más empleado para realizar estudios en enfermedades complejas.

Además de las diversas posibilidades a la hora de escoger la población, existen dos aproximaciones diferentes al realizar un estudio de asociación:

- Aproximación de genes candidatos:
    - Se tienen hipótesis causativas o genes candidatos, es decir, se basa en hipótesis previas.
    - El número de genes puede ser muy numeroso.
    - Existe una evidencia de predisposición hacia determinados genes (debido a que existen hipótesis al respecto).
    - Debido a la predisposición hacia ciertos genes, se tiene el problema de que se puede no estar estudiando los genes adecuados.
- Genome wide association studies (GWAs):
    - No necesitan una selección de genes.
    - Se consideran realmente regiones génicas, más que genes.
    - Se tiene en cuenta la densidad de SNPs / número de muestras.
    - Suelen estar basados en estudios de ligamiento.
    - Tienen la ventaja, con respecto a la aproximación anterior, de que se elimina el sesgo previo de tener unos genes que, tal vez, no son los adecuados.

2. Fundamentos computationales

La computación evolutiva (CE) se basa en realizar modelos de ciertas características de la naturaleza, fundamentalmente de la capacidad que tienen los seres vivos para adaptarse a su ambiente, lo que ya había sido tomado como base por Darwin para realizar su teoría de la evolución según el principio de selección natural en 1859 (Darwin, 1859). Las técnicas de CE se basan en este principio para desarrollar algoritmos que son capaces de adaptarse al problema que se pretende solucionar.

2.1. Algoritmos Genéticos (AA.GG.)

2.1.1. Orígenes

Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a finales de los años 60 desarrolló una técnica que permitió incorporarla en un programa de ordenador. Su objetivo era lograr que las máquinas aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" (AG) tras la publicación de su libro "Adaptation in Natural and Artificial Systems" en 1975 (Holland, 1975).

Un AG es un algoritmo de búsqueda basado en la observación de que la reproducción sexual y el principio de supervivencia del más apto permiten a las especies biológicas adaptarse a su ambiente y competir por los recursos.

Una definición bastante completa de un AG es la propuesta por John Koza: "Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando 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" (Koza, 1992).

Más formalmente, y siguiendo la definición dada por Goldberg, "Los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas" (Goldberg, 1989).

2.1.2. Codificación de las soluciones

Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos los parámetros (*genes* en la terminología de AA.GG.) se codifica en una cadena de valores denominada *cromosoma*.

El conjunto de los parámetros representado por un cromosoma particular recibe el nombre de *genotipo*. El genotipo contiene la información necesaria para la construcción del organismo, es decir, la solución real al problema, denominada *fenotipo*. Por ejemplo, en términos biológicos, la información genética contenida en el ADN de un individuo sería el genotipo, mientras que la expresión de ese ADN (el propio individuo) sería el fenotipo.

Desde los primeros trabajos de John Holland la codificación suele hacerse mediante valores binarios. Se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable representada por cada gen. El número de bits asignados dependerá del grado de ajuste que se desee alcanzar. Evidentemente no todos los parámetros tienen por qué estar codificados con el mismo número de bits. Cada uno de los bits pertenecientes a un gen suele recibir el nombre de *alelo*.

En la siguiente figura se muestra un ejemplo de un individuo binario que codifica 3 parámetros:

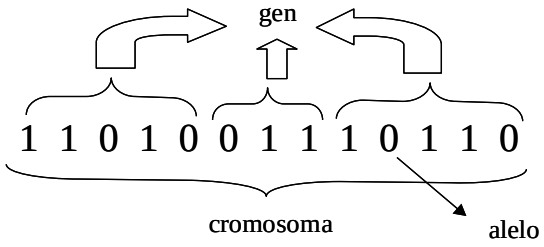


Figura 9. Individuo genético binario

Sin embargo, también existen representaciones que codifican directamente cada parámetro con un valor entero, real o en punto flotante. A pesar de que se acusa a estas representaciones de degradar el paralelismo implícito de las representaciones binarias, permiten el desarrollo de operadores genéticos más específicos al campo de aplicación del AG.

2.1.3. Funcionamiento

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida.

La evolución de estos individuos se realizará de forma análoga a como se realiza en el mundo natural: se seleccionarán un conjunto de individuos que se combinarán entre sí (cruce), y su descendencia se insertará en la población. Con una probabilidad muy baja, cuando se inserte un individuo nuevo en la población, este sufrirá una mutación. El funcionamiento general del algoritmo puede verse ilustrado en el diagrama expuesto en la Figura 10.

Los AA.GG. trabajan sobre una población de individuos. Cada uno de ellos representa una posible solución al problema que se desea resolver. Todo individuo tiene asociado un ajuste de acuerdo a la bondad con respecto al problema de la solución que representa (en la naturaleza el equivalente sería una medida de la eficiencia del individuo en la lucha por los recursos).

En la elaboración de una nueva generación, se denomina reproducción a la creación de nuevos individuos a partir de los ya existentes en la población que forma la generación anterior. Existen dos tipos fundamentales de reproducción:

- Asexual: Ocurre cuando un individuo de la generación anterior da lugar a uno de la nueva, generalmente por copia de aquél.
- Sexual: Ocurre cuando se generan varios individuos nuevos a partir del mismo número de individuos de la generación anterior. A esta operación se denomina cruce, y se basa en crear nuevas cadenas de bits a partir de subcadenas de los individuos seleccionados. Típicamente se toman dos individuos (progenitores) para generar otros dos nuevos. Se establece una posición antes de la cual los bits corresponderán a un progenitor y después de ella los bits serán los del otro. A este cruce se denomina cruce de un solo punto, pero podrían utilizarse varios puntos, correspondiendo las subcadenas de bits que separan los puntos alternativamente a progenitores distintos.

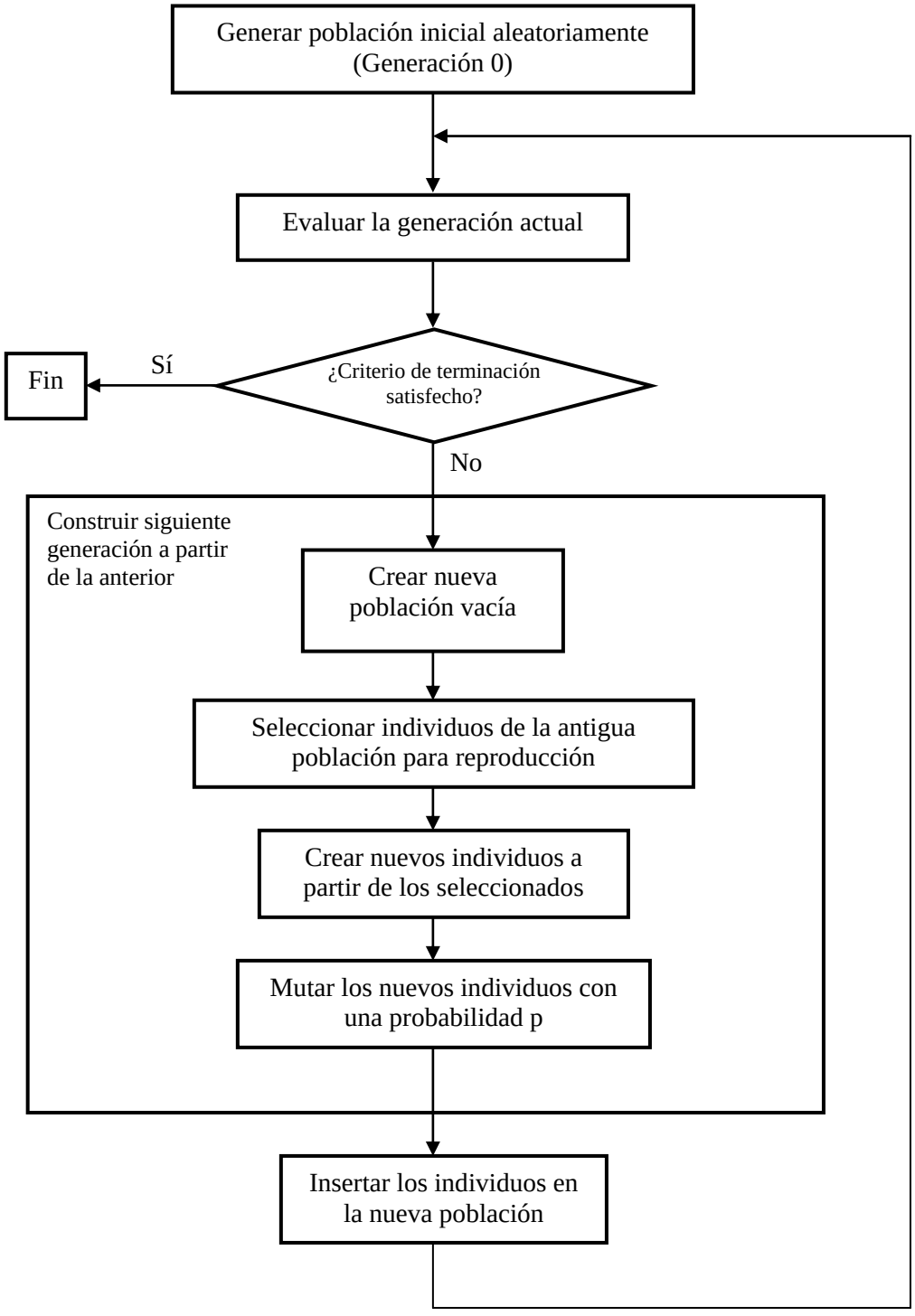


Figura 10. Diagrama de funcionamiento general del AG

Posteriormente a la creación de nuevos individuos, y con una probabilidad que suele ser muy baja (1% ~ 2%), cada individuo nuevo es sometido a un proceso de mutación: se varía la cadena de bits haciendo un cambio al azar.

Finalmente, los individuos nuevos son insertados en la nueva generación creando así una nueva población que conforma la siguiente generación.

Este proceso se repite hasta que se satisface algún criterio de parada establecido, como podrían ser:

- En la población hay algún individuo que ha alcanzado un ajuste lo suficientemente bueno.
- Ha transcurrido el número de generaciones máximo prefijado.
- La población ha convergido. Un gen ha convergido cuando el 95% de la población tiene el mismo valor para él, en el caso de trabajar con codificaciones binarias, o valores dentro de un rango especificado, en el caso de trabajar con otro tipo de codificaciones. Una vez que todos los genes alcanzan la convergencia se dice que la población ha convergido. Cuando esto ocurre la media de bondad de la población se aproxima a la bondad del mejor individuo.

Dado que cada individuo representa una posible solución al problema, la existencia de un gran número de individuos en la población implica que el algoritmo realiza una búsqueda en muchas regiones distintas del espacio de estados simultáneamente.

El resultado de la evolución (ya sea natural o simulada) no ha sido descubierto por un método de búsqueda ciego a través del espacio de estados del problema, sino por una búsqueda directa desde posiciones aleatorias en ese espacio. De hecho, de acuerdo con Goldberg (Goldberg, 1989), la evolución simulada de una solución a través de los AA.GG. en muchos casos es más eficiente y robusta que otras técnicas de búsqueda, como la ciega o la basada en cálculos.

2.1.4. Operadores genéticos

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de

no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo.

Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto, la selección de un individuo estará relacionada con su valor de ajuste. No se debe sin embargo eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea.

Para evitar una predominancia de un individuo en la reproducción, ya sea por cruce o copia, es necesario regular bien la selección de individuos para aplicarles estos operadores. Existen muchos algoritmos de selección, pertenecientes a los AA.GG. tradicionales, que realizan la tarea de escoger qué individuos se van a reproducir y cuáles no.

En general todos los algoritmos de selección se basan en la misma idea: que los individuos más aptos tengan más posibilidades de ser escogidos para reproducirse, pero sin eliminar por completo las posibilidades de los menos aptos, puesto que de ser así la población convergería en pocas generaciones. Para medir la bondad de un individuo, éste está valorado con un nivel de ajuste o aptitud, y en base a ese nivel se puede decidir la elección de individuos. Algunos de los algoritmos más utilizados son el de la selección por torneo o el de la ruleta.

En la selección por torneo (Wetzel, 1983), se escoge un número de individuos al azar de la población (típicamente dos individuos) y es seleccionado el mejor de ellos. Este método de selección es muy usado, y permite regular la presión de selección que se ejerce sobre la población variando el número de individuos que participan en el torneo. De esta forma, si participa un número bajo, se ejerce poca presión y se dan más oportunidades de ser seleccionados a los menos aptos. Conforme crece el número de individuos, serán seleccionados los mejores más frecuentemente. Un caso particular es el *elitismo global*. Se trata de un torneo en el que participan todos los individuos de la población con lo cual la selección se vuelve totalmente determinística. Además, es posible seleccionar un individuo varias veces.

Existen dos versiones de selección mediante torneo:

- ✓ Determinística.
- ✓ Probabilística.

En la versión determinística se selecciona al azar un número p de individuos (generalmente se escoge p=2). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en el paso de selección del ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio en el intervalo [0, 1], si es menor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más apto y en caso contrario el menos apto. Generalmente p toma valores en el rango $0.5 < p \leq 1$.

En la selección por ruleta (De Jong, 1975), todos los individuos de la población son dispuestos en una ruleta ocupando cada uno una parte proporcional al nivel de ajuste del individuo comparado con el nivel de ajuste de toda la población (suma de ajustes); es decir, ocupan más espacio en la ruleta los mejores individuos. Para realizar la selección, simplemente se hace girar la ruleta y se devuelve el individuo seleccionado por ella. Este método es muy utilizado por su simplicidad y sus buenos resultados. Sin embargo, presenta el problema de que al poder seleccionar el mismo individuo varias veces, el mejor individuo sea escogido muchas veces y acabe predominando en la población. Es un método muy sencillo pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$). En mucha bibliografía se suele referenciar a este método con el nombre de Selección de Montecarlo.

Existen muchos otros algoritmos, en los que el número de veces que un individuo es seleccionado se determina de forma determinística. Esto evita problemas de predominancia de un individuo. Cada uno de estos algoritmos presenta variaciones respecto al número de veces que se tomarán los mejores y peores. De esta forma, se impondrá una presión en la búsqueda en el espacio de estados en la zona donde se encuentra el mejor individuo (en el caso de que se seleccionen más veces los mejores), o bien que se tienda a repartir la búsqueda por el espacio de estados, pero sin dejar de tender a buscar en la mejor zona (caso de repartir más la selección). Los más destacados algoritmos son: "sobrante estocástico" (Booker, 1982; Brindle, 1981), "universal estocástica" (Baker, 1987) o

"muestreo determinístico" (De Jong, 1975). El auténtico motor de la búsqueda es la selección de individuos para la generación de la nueva población. La elección de un algoritmo u otro determinará la forma de búsqueda que se utilizará, estableciendo más presión de búsqueda en la mejor solución hallada hasta el momento o permitiendo la exploración de nuevas zonas en el espacio de estados.

Cruce

Una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación. Tal y como se ha indicado anteriormente el cruce es una estrategia de reproducción sexual. Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90%.

Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar). Por el contrario, utilizando una estrategia no destructiva la descendencia pasará a la siguiente generación únicamente si supera la bondad del ajuste de los progenitores (o de los individuos a reemplazar).

La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los progenitores. Al compartir las características buenas de dos individuos, la descendencia, o al menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva garantizamos que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población –aunque dispersos y posiblemente levemente modificados por la mutación–, en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente perdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación:

- ✓ Cruce de 1 punto
- ✓ Cruce de 2 puntos
- ✓ Cruce uniforme

El cruce de un punto es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera, ambos descendientes heredan información genética de los padres.

En la siguiente figura se puede ver con claridad el proceso:

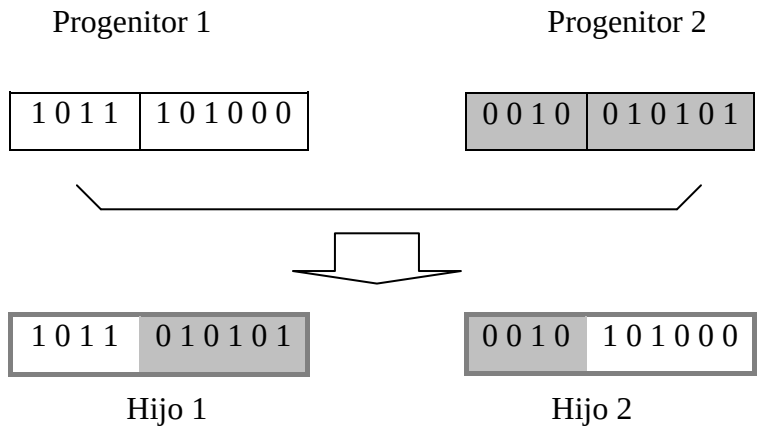


Figura 11. Ejemplo de cruce de un solo punto

En la bibliografía suele referirse a este tipo de cruce con el nombre de SPX (*Single Point Crossover*).

El cruce de dos puntos es una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres, como en el caso anterior, se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre. Generalmente se suele referir a este tipo de cruce con las siglas DPX

(Double Point Crossover). En la siguiente figura se muestra un ejemplo de cruce en dos puntos.

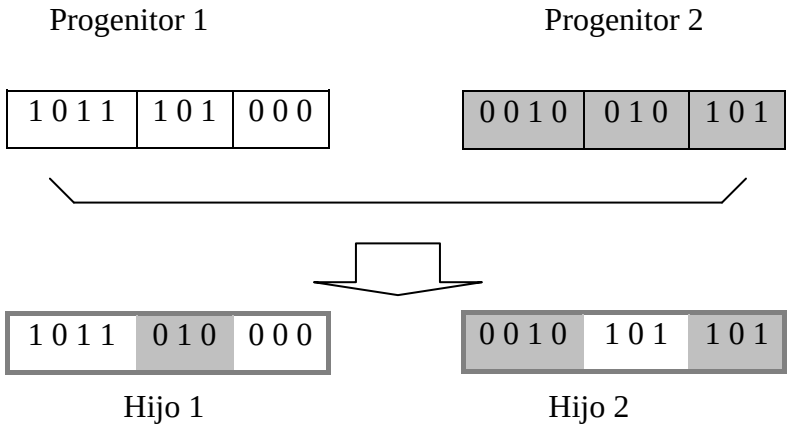


Figura 12. Ejemplo de cruce dos puntos

Generalizando, se pueden añadir más puntos de cruce dando lugar a algoritmos de cruce multipunto. Sin embargo existen estudios que desaprueban esta técnica (De Jong, 1975). Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora con respecto al cruce de un solo punto, el hecho de añadir un mayor número de puntos de cruce reduce el rendimiento del AG. El problema principal de añadir nuevos puntos de cruce radica en que es más fácil que los segmentos originados sean corrompibles; es decir, que por separado quizás pierdan las características de bondad que poseían conjuntamente. Sin embargo, no todo son desventajas y añadiendo más puntos de cruce se consigue que el espacio de búsqueda del problema sea explorado más a fondo.

El cruce uniforme, en cambio, es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro progenitor. Aunque se puede implementar de muy diversas formas, la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer progenitor. Si por el contrario hay un 0 el gen se copia del segundo progenitor. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

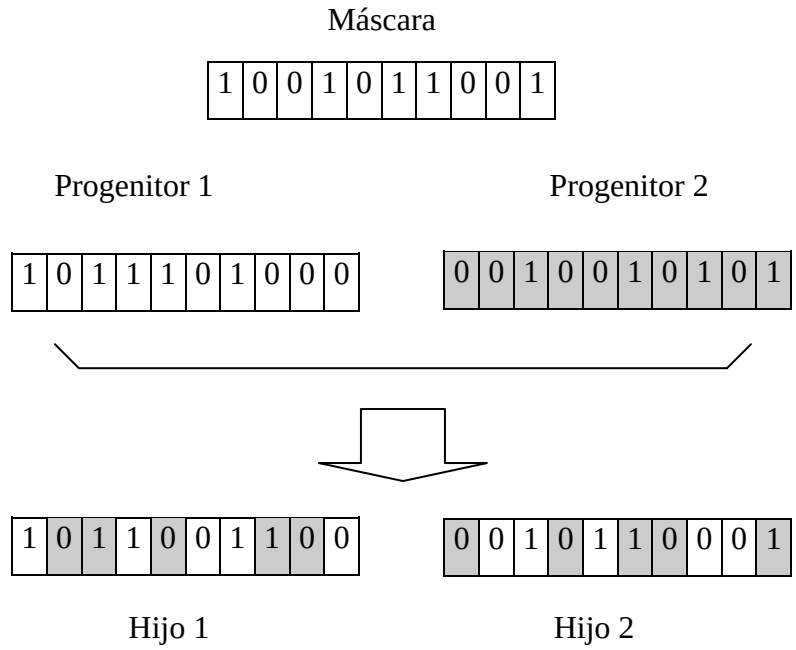


Figura 13. Ejemplo de cruce uniforme

Tal y como se puede apreciar en la figura anterior, la descendencia contiene una mezcla de genes de cada uno de los progenitores. El número efectivo de puntos de cruce no es fijo, pero será por término medio L/2, siendo L la longitud del cromosoma (número de alelos en representaciones binarias o de genes en otro tipo de representaciones).

La máscara de cruce no permanece fija durante todo el proceso evolutivo. Se genera de manera aleatoria para cada cruce.

Se suele referir a este tipo de cruce con las siglas UPX (*Uniform Point Crossover*).

Los tres tipos de cruce vistos hasta el momento son válidos para cualquier tipo de representación del genotipo. Si se emplean genotipos compuestos por valores enteros o reales pueden definirse otros tipos de operadores de cruce:

- Media: el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos progenitores.
- Media geométrica: cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.

- Extensión: se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma al valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente.

Copia

La copia es otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera, el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce, y si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

Mutación

La mutación de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se suele utilizar de manera conjunta con el operador de cruce. Primero se seleccionan dos individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad *Pm*. Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La probabilidad de mutación es muy baja, generalmente menor al 1%. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo, se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser examinado.

Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Este consiste en variar aleatoriamente un gen de un cromosoma. Si se trabaja con codificaciones binarias consistirá simplemente en negar un bit. También es posible realizar la mutación intercambiando los valores de dos alelos del cromosoma. Con otro tipo de codificaciones no binarias existen otras opciones:

- ✓ Incrementar o decrementar a un gen una pequeña cantidad generada aleatoriamente.
- ✓ Multiplicar un gen por un valor aleatorio próximo a 1.

Aunque no es lo más común, existen implementaciones de AA.GG. en las que no todos los individuos tienen los cromosomas de la misma longitud. Esto implica que no todos ellos codifican el mismo conjunto de variables. En este caso existen mutaciones adicionales como puede ser añadir un nuevo gen o eliminar uno ya existente.

Algoritmos de reemplazo

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo:

- Aleatorio: el nuevo individuo se inserta en un lugar cualquiera de la población.
- Reemplazo de padres: se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.
- Reemplazo de similares: una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.
- Reemplazo de los peores: de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

2.1.5. Evaluación

Para el correcto funcionamiento de un AG se debe de poseer un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto, para cada tipo de problema que se desee resolver deberá derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos.

De esto se encarga la función de evaluación, también denominada función de fitness, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste o fitness. En la naturaleza el ajuste (o adecuación) de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad deberá estar ponderada con el número de descendientes. Evidentemente no es lo mismo una probabilidad de reproducción del 25% en una población de un par de cientos de individuos que esa misma probabilidad en una población de varios millones.

En el mundo de los AA.GG. se empleará esta medición para controlar la aplicación de los operadores genéticos. Es decir, permitirá controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo.

La cuantificación de la bondad de un determinado individuo se realiza por medio del ajuste de ese individuo. Este valor representa lo bien que el fenotipo del individuo soluciona el problema actual.

El ajuste es probablemente el principal concepto en la evolución darwiniana, se refiere a la habilidad que tiene un individuo de competir en un entorno por los recursos disponibles. Goldberg describió la función de ajuste como "*una medida de beneficio, utilidad o bondad que queremos maximizar*" (Goldberg, 1989).

En el AG, esta competición se basa en la actuación del cromosoma dentro del dominio del problema. Se determina una escala adecuada a la tarea como "tiempo antes de fallo" (Randall et al., 1994), o "tiempo antes de estabilizarse" (Koza, 1990). Después de haber aplicado un cromosoma al problema, se le asigna un valor de ajuste que refleje su actuación. De esta manera, cuando la población entera haya sido probada, la habilidad relativa de cada cromosoma puede ser identificada.

Para valorar esta medida de ajuste, existen tres tipos fundamentales (Koza, 1992):

- Estandarizado. Este tipo de ajuste s(i,t) mide la bondad de un individuo i en la generación t, de tal forma que valores próximos a cero indican un buen valor de ajuste y valores lejanos un mal individuo. Por tanto, en una generación t un individuo i será peor que otro j si s(i,t)>s(j,t). Esta medida es muy útil en problemas en los que la cuantificación del nivel de ajuste de los individuos se basa en penalizaciones, como puede ser el error en inducción de fórmulas, error cuadrático medio, número de ejecuciones necesarias para encontrar la solución, etc.

- Ajustado. Este valor se obtiene a partir del ajuste estandarizado de la siguiente forma:

$$a(i,t) = \frac{1}{1 + s(i,t)}$$

  Con esta medición la bondad se cuantifica entre 0 y 1, siendo el valor 1 correspondiente al mejor individuo.

- Normalizado. Es un valor de ajuste comparativo del individuo con toda la población. Se obtiene de la siguiente expresión, dado el tamaño de población M:

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{M} a(k,t)}$$

  Este valor está comprendido entre 0 y 1. Este tipo de ajuste indica el nivel de bondad dentro de la población: ha desaparecido el componente de objetividad de evaluación de los tipos anteriores y un valor cercano a 1 ya no indica que ese individuo represente una solución buena al problema, sino que ese individuo representa una solución destacada y notablemente mejor que la del resto de la población. Este valor es utilizado para las selecciones proporcionales al ajuste, como la ruleta. En este caso, la proporción de ruleta ocupada por un individuo será este valor, ya que la suma de todos será la unidad.

La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se les asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido. Tal y como se ha comentado, este procedimiento de evaluación será específico del dominio del problema en el que se aplica el AG. También puede calcularse el ajuste mediante una manera "co-evolutiva". Por ejemplo, el ajuste de una estrategia de juego se determina aplicando esa

estrategia contra la población entera (o en su defecto una muestra) de estrategias de oposición.

2.1.6. Parámetros

Los principales parámetros que se pueden configurar para variar la ejecución son:

- Tamaño de la población. Este parámetro indica el número de individuos que va a tener la población. En general este parámetro se ajusta de forma proporcional a la complejidad del problema, tomando valores altos cuanto mayor sea ésta. De esta forma, cuanto más complicado es un problema, habrá más opciones de conseguir mejores resultados en un menor número de generaciones, puesto que se generan más individuos nuevos. Sin embargo, un tamaño alto puede no ser siempre la mejor solución: es posible tomar un tamaño menor y confiar más en la evolución durante mayor número de generaciones (Fuchs, 1999; Gathercole and Ross, 1997).
- La tasa de cruces, es decir, el porcentaje de individuos de la siguiente generación que serán creados a partir de cruces de individuos de la anterior. Esta tasa suele ser alta, generalmente supera el 90%.
- La probabilidad de mutación. Esta probabilidad suele ser muy baja (menor de 0.05).
- Los algoritmos de cruce, selección y mutación utilizados.
- El criterio de parada del algoritmo. Como ya se ha explicado, este puede ser:
    - Número máximo de generaciones.
    - Haber alcanzado un valor de ajuste aceptable.
    - Convergencia de la población.

2.2. Programación Genética (PG)

2.2.1. Orígenes

La programación genética surge como una evolución de los algoritmos genéticos tradicionales, manteniendo el mismo principio de selección natural. Lo que ahora se pretende es resolver los problemas mediante la inducción de programas y algoritmos.

Ya en las primeras conferencia sobre algoritmos genéticos, la primera desarrollada en la Universidad de Carnegie Mellon en 1985, y la segunda en Hillsdale en 1987, se puede encontrar dos artículos que, sin usar explícitamente el nombre de programación genética (que fue introducido por John R. Koza), pueden ser considerados como precursores en la materia: "A Representation for the Adaptive Generation of Simple Sequential Programs" (Cramer, 1985) y "Using the Genetic Algorithm to Generate Lisp Source Code to solve the Prisoner's Dilemma" (Fujiki and Dickinson, 1987).

El primero de estos artículos plantea un sistema adaptativo para la generación de pequeños programas secuenciales. Para ello, hace uso de dos lenguajes: JB (que representa los programas en forma de cadenas de números) y TB (versión evolucionada de JB, pero con estructura de árbol para representar programas). El objetivo principal del artículo es conseguir una forma de representar programas que, por un lado, permita la aplicación de los operadores genéticos clásicos (mutación, cruce, inversión) y que, por otro lado, produzca sólo programas "bien formados", incluso cuando se apliquen dichos operadores sobre los programas. No es importante que todos los programas que se puedan obtener sean útiles (de eso ya se encargarán los criterios de selección): sólo importa que estén dentro del espacio de programas sintácticamente correctos.

La gran importancia de este artículo se halla en la constatación de la importancia que tiene la representación de los programas para su manipulación. Los problemas que plantea el lenguaje JB pueden ser eliminados si se usa TB y su estructuración en forma de árboles. Este hecho, que en principio puede ser considerado poco relevante, ha demostrado ser de gran utilidad en trabajos posteriores, y, de hecho, todo el material actual que hay sobre programación genética se basa en la representación arbórea.

El segundo artículo trata sobre la resolución de un problema clásico: el dilema del prisionero. En él, dos prisioneros son interrogados por separado. Cada uno debe decidir si delatar o no al otro, y en función de la declaración de ambos obtienen puntos. Si ambos delatan al otro, obtienen cada uno 1 punto. Si ninguno delata al otro, obtienen cada uno 3 puntos. Si uno delata y el otro no, el delator obtiene 4 puntos y el otro 0. Al cabo de un número determinado de interrogatorios (rondas) el que obtenga más puntos gana. Este problema ha sido usado como modelo en multitud de situaciones, desde el nivel de las relaciones personales hasta las negociaciones entre grandes corporaciones.

El interés del artículo se centra en el apartado dedicado a la representación del conocimiento. En teoría, el sistema debería ser lo suficientemente flexible como para encontrar una solución única y novedosa, aunque en la práctica suele ser necesario, e incluso deseable, proporcionar cierto tipo de conocimiento sobre la solución para agilizar el proceso de búsqueda. En el sistema de Hicklin, por ejemplo, este conocimiento es representado en forma de programa, generado por medio de una serie de producciones. El grado de flexibilidad del sistema y el grado de conocimiento incorporado dependerá en cómo sean definidas dichas producciones. Para el artículo, en concreto, se usó un conjunto restringido de expresiones LISP, que demostró ser de gran utilidad y funcionalidad.

2.2.2. Codificación de programas

La mayor diferencia entre los AA.GG. y la PG es la forma de codificación de la solución al problema. En PG la codificación se realiza en forma de árbol, de forma similar a como los compiladores leen los programas según una gramática prefijada.

Elementos del árbol

Al haber una representación de árbol, existirán dos tipos de nodos:

- Terminales, u hojas del árbol. Son aquellos que no tienen hijos. Normalmente se asocian con valores constantes o variables.
- Funciones. Son aquellos que tienen uno o más hijos. Generalmente se asocian con operadores del algoritmo que se quiere desarrollar.

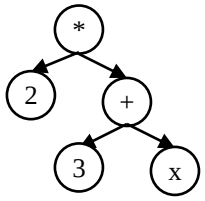


Figura 14. Árbol para la expresión 2*(3+x)

En la figura anterior se puede ver un ejemplo de árbol, que representa el programa f(x) = 2*(3+x). Tiene como nodos no terminales los correspondientes al producto y a la suma, y como terminales los correspondientes a los valores 2 y 3 y la variable x.

Una parte fundamental del funcionamiento de la programación genética es la especificación del conjunto de elementos terminales y funciones antes del inicio del proceso evolutivo. Con los nodos que se le especifique, el algoritmo construirá los árboles. Por tanto, es necesario un mínimo proceso de análisis del problema para configurar el algoritmo, puesto que hay que decirle qué operadores puede utilizar. Como regla general, es conveniente ajustar el número de operadores sólo a los necesarios, puesto que la adición de elementos que no sean necesarios no provocará que no se encuentre la solución, pero sí que el algoritmo tarde más en encontrarla.

A la hora de especificar los conjuntos de elementos terminales y funciones, es necesario que estos conjuntos posean dos requisitos, que son *cerradura* y *suficiencia*. El requisito de suficiencia dice que la solución al problema debe poder ser especificada con el conjunto de operadores especificados. El requisito de cerradura dice que debe ser posible construir árboles correctos con los operadores especificados.

Dado que el proceso del construcción de árboles es un proceso basado en el azar, muchos de los árboles construidos no serán correctos no por no seguir las reglas de la gramática sino por la aplicación de operadores (nodos no terminales) a elementos que no están en su dominio. Por esta razón no se aplican estos operadores directamente, sino una modificación de ellos en la que se amplía su dominio de aplicación. El ejemplo más claro es el operador de división, cuyo dominio es el conjunto de números reales excepto el valor cero. Ampliando su dominio, se define un nuevo operador (%):

$$\%(a,b) = \begin{cases} 1, si\ b = 0 \\ a/b\ , si\ b \neq 0 \end{cases}$$

A esta nueva operación se denomina *operación de división protegida*, y, en general, cuando se crea una nueva operación, que extiende el dominio de otra, se denomina *operación protegida*.

Restricciones

Existen dos tipos principales de restricciones:

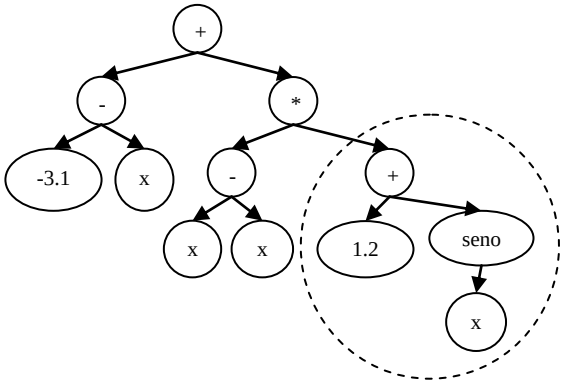
- ✓ Tipado.
- ✓ Altura máxima del árbol.

Para establecer reglas sintácticas en la creación de árboles, es posible especificar reglas de tipado (Montana, 1995): se establece el tipo de cada nodo (terminales y funciones), y para los no terminales (funciones) el tipo que debe tener cada hijo. De esta forma se especifica la estructura que deben seguir los árboles.

Al especificar el tipo de cada nodo, se está especificando una gramática que va a ser la que siga el algoritmo para la construcción de árboles. Esta gramática permitirá que los árboles tengan la estructura deseada.

Los tipos más usados son aquellos que tienen que ver con la realización de operaciones aritméticas: reales y enteros. Sin embargo, son también muy usados otros como el tipo booleano o el tipo sentencia. Este último se utiliza cuando se quiere desarrollar un programa que sea una secuencia de mandatos y designa nodos que no devuelven nada, sino que su evaluación se basa en los efectos laterales que provoca su ejecución.

La restricción de altura evita la creación de árboles demasiado grandes y fuerza una búsqueda en soluciones cuyo tamaño se acota de antemano. Con esta restricción se evita que los árboles posean mucho código redundante y el crecimiento excesivo de los árboles (Soule, 1998; Soule and Foster, 1997). El crecimiento excesivo de los árboles es un fenómeno conocido con el nombre de *bloat*. Este fenómeno se produce de forma espontánea al avanzar el proceso evolutivo, puesto que los árboles evolucionan generando partes que no influyen en su comportamiento. Esto se produce para paliar los efectos nocivos de los operadores de cruce y mutación, puesto que cuantas más partes inútiles tenga un árbol, menos probabilidades habrá de que este sea modificado cuando se le aplique un operador de cruce o mutación. De esta forma, los individuos se protegen a sí mismos. Un ejemplo de una parte inútil de un árbol sería, en un individuo que representa una expresión matemática, un subárbol al cual se va a multiplicar por cero, como se puede ver en la Figura 15.



-3.1 – x + ((x - x) * (1.2 + seno(x)))

Figura 15. Ejemplo de árbol con una rama inútil

2.2.3. Funcionamiento

Algoritmo principal

El funcionamiento es similar al de los algoritmos genéticos: se basa en la generación de sucesivas generaciones a partir de las anteriores. Este algoritmo se puede ver en la Figura 16 (Koza, 1992).

Tras la creación inicial de árboles (generalmente serán aleatorios), se construyen sucesivas generaciones a partir de copias, cruces y mutaciones de los individuos de cada generación anterior.

Para poder aplicar la PG será necesario especificar dos elementos fundamentales:

- ✓ Conjunto de terminales y funciones. Dada la forma diferente de codificación que tiene la PG frente a los AA.GG., será necesario especificar qué elementos se pueden utilizar para construir los árboles.
- ✓ Función de ajuste. Indica la bondad de cada individuo.

Generación inicial de árboles

El primer paso en el funcionamiento del algoritmo es la generación de la población inicial. En la creación de la generación 0, cada árbol se creará de forma más o menos aleatoria, dependiendo del algoritmo, teniendo en cuenta las restricciones que existen en los árboles.

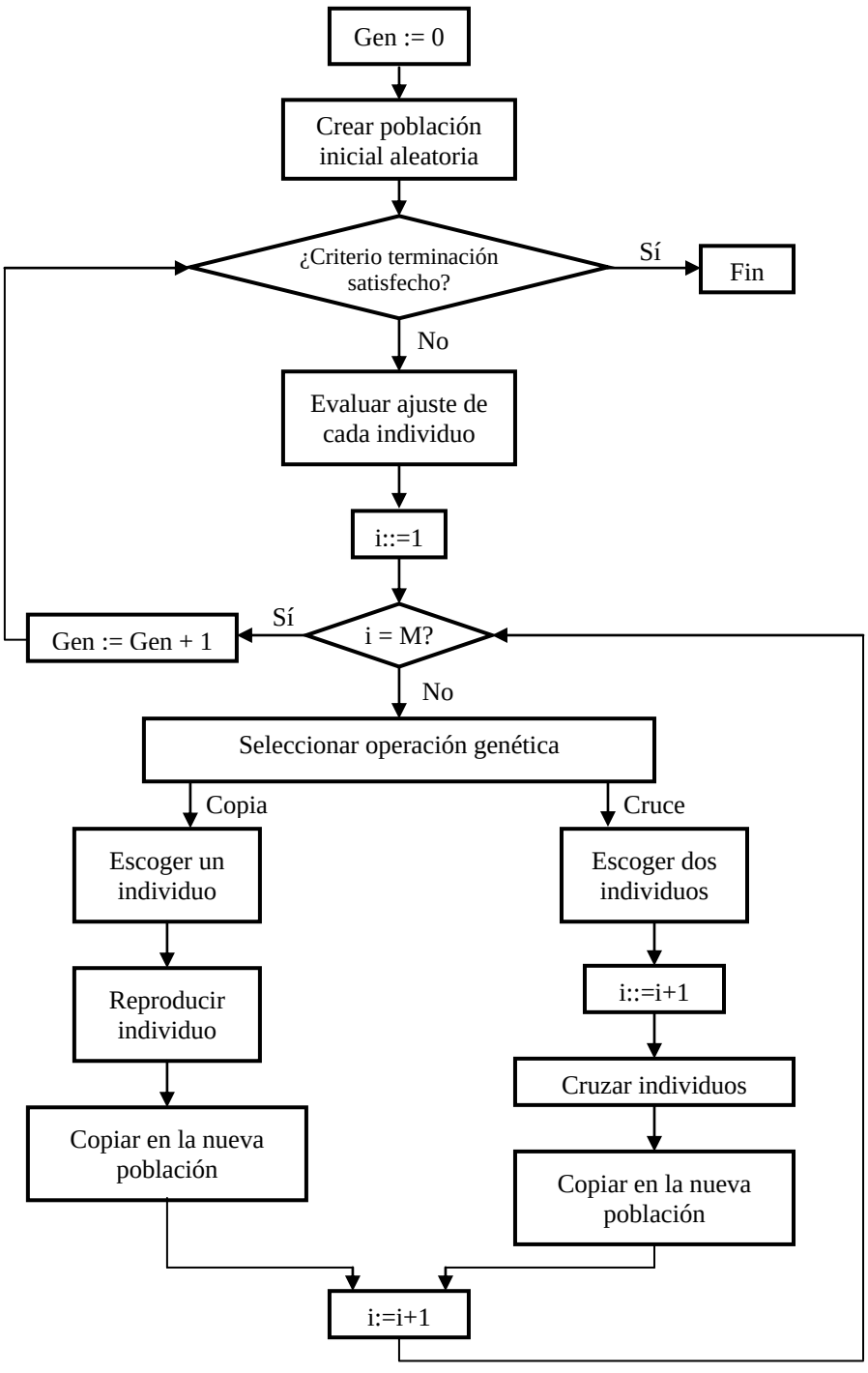


Figura 16. Diagrama de flujo de programación genética

Dado que los árboles son aleatorios, los individuos de esta población en general representan soluciones malas al problema.

Para la creación de un árbol existe una gran variedad de algoritmos, pero son tres los más utilizados (Koza, 1992): parcial, completo e intermedio.

El algoritmo de creación parcial genera árboles cuya altura máxima no supera la especificada. El algoritmo es el siguiente, dada la altura máxima y los conjuntos de elementos terminales (T) y funciones (F):

```
GeneraÁrbol (altura, F, T)
begin
    if altura=1 then
            Asignar como raíz del árbol un elemento aleatorio de T
    else
            Asignar como raíz del árbol un elemento aleatorio de F∪T
    Para cada hijo de la raíz,
        Asignar a la raíz como hijo el subárbol generado con
    GeneraÁrbol (altura-1, F, T)
    end
```

En este algoritmo, cada hoja tendrá como máximo la profundidad especificada. Para el caso de generar árboles tipados, será necesario mantener la restricción del tipado, con lo que el algoritmo es el siguiente:

```
GeneraÁrbol (altura, F, T, tipo)
    begin
            if altura=1 then
                Asignar la raíz del árbol como un elemento aleatorio de T
                    del tipo especificado.
            else
                    Asignar la raíz del árbol como un elemento aleatorio de
                            F∪T del tipo especificado.
            Para cada hijo de la raíz,
                Asignar a la raíz como hijo el subárbol generado con
                    GeneraÁrbol (altura-1, F, T, tipo de ese hijo)
    end
```

El algoritmo completo genera árboles cuyas hojas están todas a un determinado nivel, pues genera árboles completos. El algoritmo es muy similar al anterior:

```
GeneraÁrbol (altura, F, T)
    begin
        if altura=1 then
            Asignar la raíz del árbol como un elemento aleatorio de T
        else
        Asignar la raíz del árbol como un elemento aleatorio de F
        Para cada hijo de la raíz,
            Asignar a la raíz como hijo el subárbol generado con
                GeneraÁrbol (altura-1, F, T)
    end
```

Para el caso de utilizar las propiedades de tipado, se realiza la modificación como anteriormente, dando lugar a:

```
GeneraÁrbol (altura, F, T, tipo)
    begin
        if altura=1 then
            Asignar la raíz del árbol como un elemento aleatorio de T
                del tipo especificado.
        else
        Asignar la raíz del árbol como un elemento aleatorio de F
                del tipo especificado.
        Para cada hijo de la raíz,
            Asignar a la raíz como hijo el subárbol generado con
                GeneraÁrbol (altura-1, F, T, tipo de ese hijo)
    end
```

Sin embargo, es necesario tener en cuenta que al utilizar tipado se está forzando a la utilización de un conjunto de reglas sintácticas y esto puede provocar que sea imposible generar árboles de la altura especificada con los conjuntos de terminales y funciones dados.

Por ejemplo, con el conjunto de terminales T y funciones F

$$T = \{ X, Y, 3, 5 \} \qquad F = \{ \text{if}, > \}$$

Sería imposible generar un árbol completo de tipo real de altura 2, puesto que para ello la raíz debería ser el elemento "if", con su primer hijo de tipo booleano y como operador booleano solo hay un nodo no terminal, el que representa la relación de mayor, y se necesita un nodo terminal para garantizar el cumplimiento de la restricción de altura. En este ejemplo los conjuntos F y T cumplen la condición de cerradura, pues es posible construir árboles correctos, pero existe una incompatibilidad en el algoritmo completo con la restricción de altura 2 al generar árboles iniciales causada por la gramática que se está implementando.

El algoritmo de creación intermedio es una mezcla de los dos anteriores, creado para que exista mayor variedad en la población inicial, y con ello mayor diversidad genética. Este algoritmo se basa en ejecutar los anteriores alternándolos y tomando distintas alturas para crear todos los elementos de la población. El algoritmo es el siguiente: dado un tamaño de población M y una altura máxima A:

```
for i:=2 to A do begin
    Generar M/(2*(A-1)) árboles de altura i con el método parcial
    Generar M/(2*(A-1)) árboles de altura i con el método completo
end
```

Este método genera un porcentaje de 100/(A-1)% árboles nuevos de altura, variando entre 2 y A, de forma alternativa, completos y parciales.

En general, se evita que se generen árboles de altura 1; es decir, árboles que contengan solo un elemento terminal. En la práctica se modifican los algoritmos para que se evite esta posibilidad.

Estos algoritmos se basan fuertemente en el azar, y la única intervención del usuario está en la introducción de los elementos terminales y funciones. Sin embargo, existen muchos más algoritmos en los que la creación de árboles no es tan aleatoria. Por ejemplo, en (Luke, 2000) se asigna una probabilidad de aparición a cada nodo y, de esta forma, se reduce el carácter aleatorio de la creación. Además, se orientan los árboles a que contengan más nodos de una clase que otra. Los algoritmos serán muy similares, con la salvedad de que la elección de los elementos seguirá siendo aleatoria, pero estará ponderada por esa probabilidad asignada.

2.2.4. Operadores genéticos

En la creación de una nueva generación se aplican operadores en los que se generan y modifican los árboles. Estos operadores son resultado de adaptar los existentes en los algoritmos genéticos tradicionales a la programación genética, modificándolos para adaptarlos a la codificación en forma de árbol.

Los operadores más utilizados son:

- ✓ Cruce.
- ✓ Reproducción.
- ✓ Selección.
- ✓ Mutación.

Cruce

El principal operador es el de cruce. En él, dos individuos de la antigua población se combinan para crear otros dos individuos nuevos.

Después de seleccionar a dos individuos como padres, se selecciona un nodo al azar en el primero y otro en el segundo de forma que su intercambio no viole ninguna de las restricciones: los nodos deben ser de igual tipo y los árboles nuevos deben seguir manteniendo la altura máxima. El cruce entre los dos padres se efectúa mediante el intercambio de los subárboles seleccionados en ambos padres.

La restricción de altura conlleva a que si se está tomando una altura máxima de A y se selecciona un nodo de un árbol para cruce que está a una profundidad P, del segundo árbol se descartarán para el cruce todos aquellos nodos cuyos subárboles tengan una altura mayor que A-P, puesto que el resultado del cruce daría lugar a un árbol de altura mayor que A. De la misma forma, si el nodo seleccionado en el primer árbol representa a un subárbol de altura A', del segundo se descartan aquellos cuya profundidad sea mayor que A-A', puesto que de insertar el subárbol seleccionado del primer árbol en ese hueco, se violaría igualmente la restricción de altura máxima.

La Figura 17 muestra un ejemplo de dos árboles seleccionados para cruce. En este caso, la altura máxima que se utiliza es de 5.

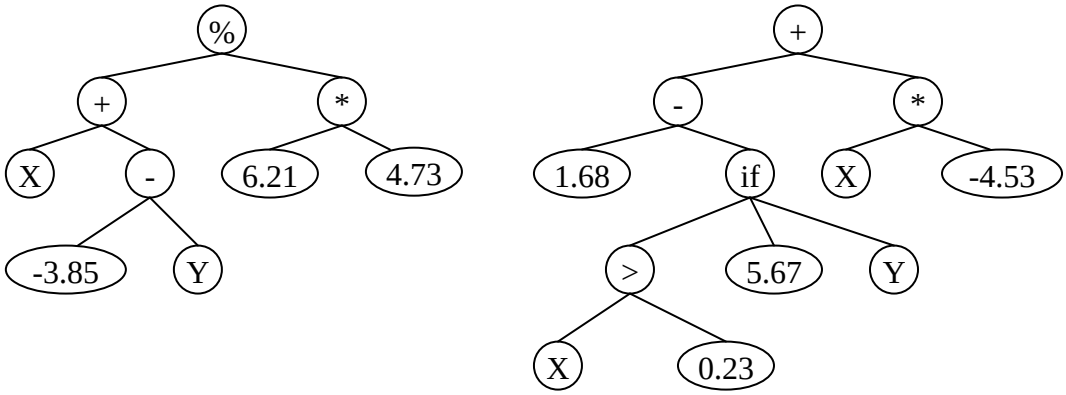


Figura 17. Árboles seleccionados para cruce

En primer lugar, se selecciona de forma aleatoria un nodo en el primer árbol. En este caso el nodo "+", que representa a un subárbol de altura 3. En el segundo árbol se descarta aquellos nodos de tipo distinto, en este caso el nodo ">", por ser de tipo booleano (Figura 18).

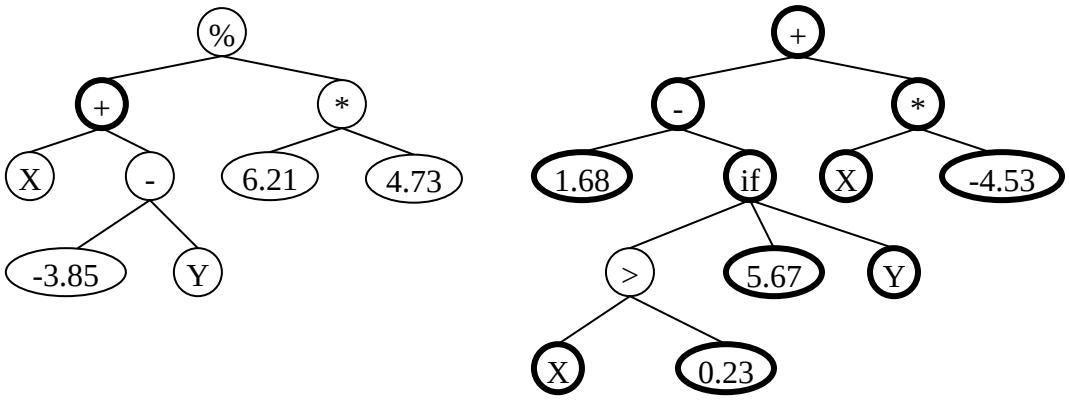


Figura 18. Se selecciona nodo en el primer progenitor y se aplica restricción de tipo en el segundo

En el segundo árbol se descartan aquellos nodos que nos llevarían a violar la restricción de altura tras el intercambio. Se descarta la raíz del árbol porque llevaría a un árbol demasiado alto en el primer progenitor y los nodos de altura 4 y 5 (es decir, los dos últimos niveles) porque llevarían a un árbol demasiado alto en el segundo progenitor (Figura 19).

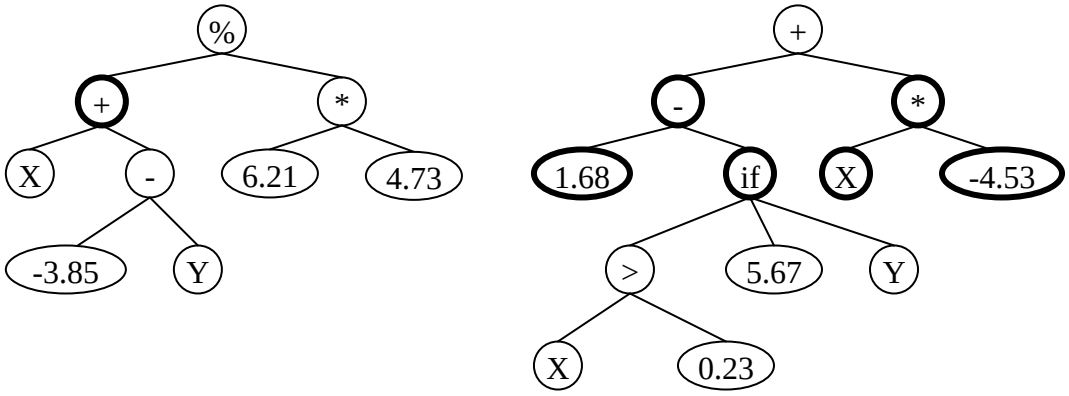


Figura 19. Se descartan nodos en el segundo progenitor que violen la restricción de altura máxima

En el segundo árbol se selecciona un nodo de los restantes, en este caso el operador "if", y se intercambian los nodos (Figura 20 y Figura 21).

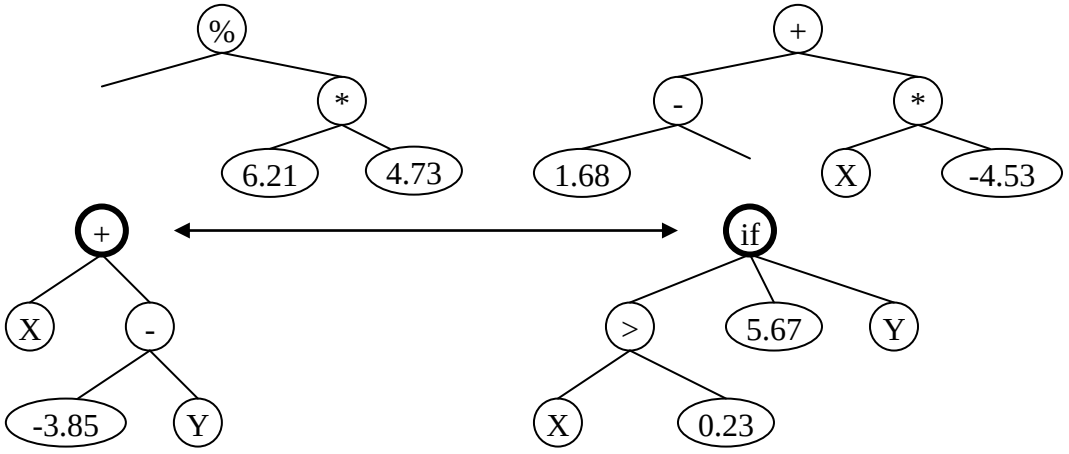


Figura 20. Se selecciona un nodo de los restantes

La operación de cruce es sexual en el sentido en que se necesitan dos individuos para generar individuos nuevos.

Este operador es el auténtico motor del funcionamiento de la programación genética y provoca la combinación de resolución de subproblemas (cada subárbol puede interpretarse como una forma de resolver un subproblema) para la resolución del problema principal.

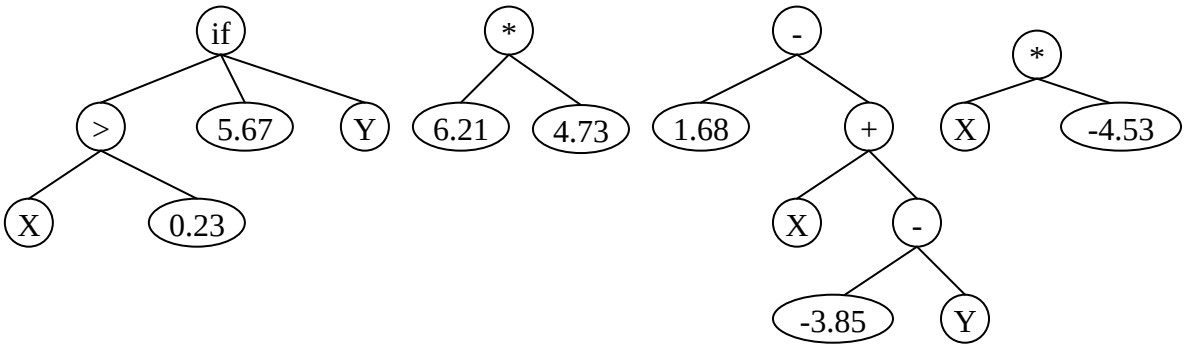


Figura 21. Se intercambian los subárboles y se introducen en la nueva población

Se ha observado (Soule, 1998) que la mayoría de los cruces provocan la generación de individuos peores, así como la aparición de mucho código redundante dentro de los árboles. Este código redundante previene los posibles efectos nocivos de otros operadores más destructivos como el de mutación, pero provoca un crecimiento exagerado de los árboles en poco tiempo. Por ello, una solución es utilizar *cruces no destructivos*: los árboles generados por la operación de cruce son insertados en la nueva generación si son mejores que sus padres. En caso contrario, se insertan en la nueva generación copias de los padres.

Una ventaja de este tipo de cruces sobre los cruces de los algoritmos genéticos tradicionales es que al cruzar dos padres iguales, los hijos en general son distintos a los padres (y distintos entre ellos). Esto no ocurría en el cruce en los algoritmos genéticos, en los que en este caso, cuando se cruzaban padres idénticos, los hijos eran iguales a los padres.

Por ejemplo, si se cruzan los nodos señalados de los árboles iguales presentes en la Figura 22.

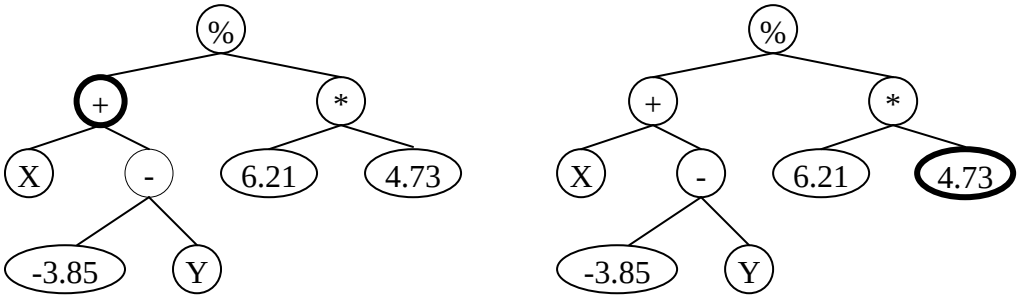


Figura 22. Árboles iguales como ejemplo de cruce

En este caso, se obtienen los árboles de la Figura 23, que son diferentes.

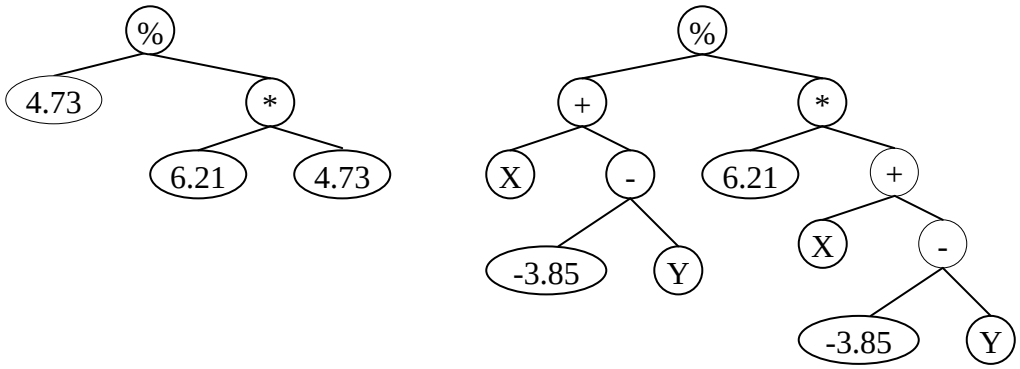


Figura 23. Resultado de realizar un cruce entre árboles iguales

Al realizar un cruce, el nodo escogido en ambos árboles no se suele tomar al azar: generalmente se asigna una probabilidad de que el nodo tomado sea no terminal. Esta probabilidad suele ser alta (sobre 0.9), puesto que en un árbol la mayoría de los nodos son terminales, y si no se toma esta probabilidad la mayoría de los cruces no son más que permutaciones de elementos terminales entre distintos árboles.

El operador de cruce aquí descrito se realiza de forma igual para todos los individuos. Sin embargo, existe variantes adaptativas de estos operadores (Angeline, 1996) en las que el propio algoritmo se modifica, así como numerosas variantes que tienen como base este algoritmo (Aguirre et al., 1999; Pereira et al., 1999).

Reproducción

La reproducción simplemente es la copia de individuos en la nueva generación. Esta operación es asexual en el sentido en que se genera un individuo a partir de un individuo anterior.

Esta operación, junto con la de cruce, son las que se utilizan más frecuentemente, y entre ellas tiene una especial predominancia la de cruce. De hecho, el porcentaje de individuos nuevos generados a partir de cruces suele ser superior al 90%, mientras que el resto son generados mediante copias. El aumento del número de individuos generados por copias aumenta el peligro de predominancia de un individuo sobre el resto de la población, y que finalmente tras varias generaciones toda la población converja hacia ese individuo. Esta es una situación indeseable, puesto que se ha perdido por completo la diversidad genética que

se tenía al principio y ello conlleva que la búsqueda en el espacio de estados sólo se lleve a cabo en una determinada zona, lo cual es lo contrario que se pretende.

El operador de cruce es por tanto el principal operador utilizado para la generación de los nuevos árboles y exploración del espacio de estados (Poli and Langdon, 1998).

Selección

Al igual que en el caso de los AA.GG., la selección de individuos debe de realizarse favoreciendo el escoger individuos sobre todo de los mejores de la población, pero permitiendo también que algunos de los peores sean escogidos. Dado que en esta operación no influye la forma de codificación de los individuos, los algoritmos que se utilizan son los mismos que para el caso de AA.GG., explicados anteriormente.

Mutación

El operador de mutación provoca la variación de un árbol de la población. Este operador suele usarse con probabilidad muy baja (menos que 0,1) antes de introducir un individuo en la nueva generación.

Existen dos tipos principales de mutación: mutación en la que se varía un solo nodo y mutación en la que se varía una rama entera del árbol.

En el primer caso, conocida por mutación puntual, la mutación actúa de la siguiente manera:

1. Se escoge un nodo al azar del árbol.
2. Se escoge al azar un nodo del conjunto de terminales o no terminales, del mismo tipo que el seleccionado, con el mismo número de hijos y de forma que sus hijos sean del mismo tipo.
3. Se intercambia el nodo antiguo del árbol por el nuevo, manteniendo los mismos hijos que el antiguo.
4. Dado que cada rama del árbol representa una solución a un subproblema y el no terminal que las une representa la forma de combinar esas soluciones, si se realiza este tipo de mutación sobre un elemento no terminal se estará provocando que las soluciones se combinen de distinta forma. Este tipo de mutación apenas se usa. La Figura 24 ilustra un ejemplo de mutación de este tipo. En este ejemplo, se ha

mutado el árbol cambiando el nodo destacado por otro del mismo tipo y mismo número de hijos.

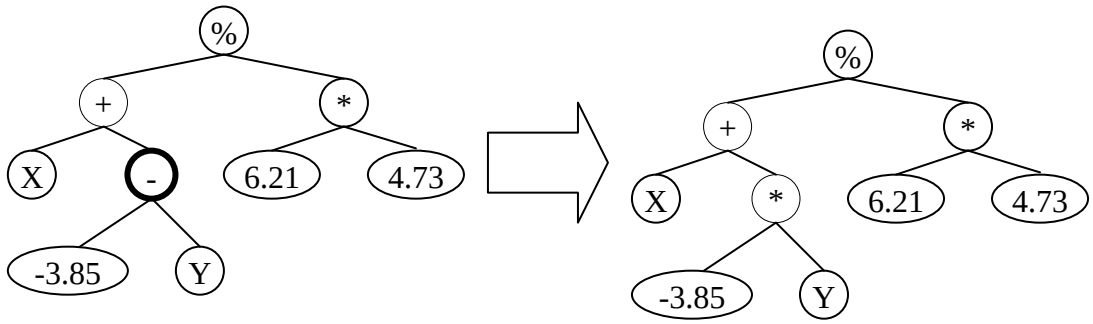


Figura 24. Ejemplo de mutación puntual

En el segundo tipo de mutación, se efectúan cambios mayores en el árbol. La operación es la siguiente: se escoge un nodo al azar del árbol, se elimina todo el subárbol que cuelga de ese nodo, se crea un nuevo subárbol del tipo y altura adecuados y se pone en su lugar.

Al cambiar una rama entera, lo que ahora se cambia del árbol es la forma de resolver el subproblema. En la Figura 25 puede verse un ejemplo de un árbol sobre el que se va a aplicar esta operación de mutación. En él puede apreciarse un nodo destacado, que será el que sufra el proceso de mutación.

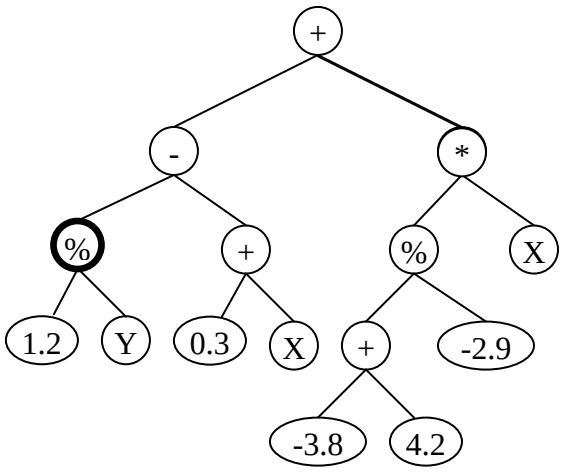


Figura 25. Árbol sobre el que se realizará mutación

Para realizar la mutación, será necesario eliminar el nodo seleccionado (y el subárbol que representa) y crear un subárbol nuevo para insertarlo en ese lugar. Para no violar la restricción de altura, el subárbol nuevo deberá tener una altura máxima de 3 (Figura 26).

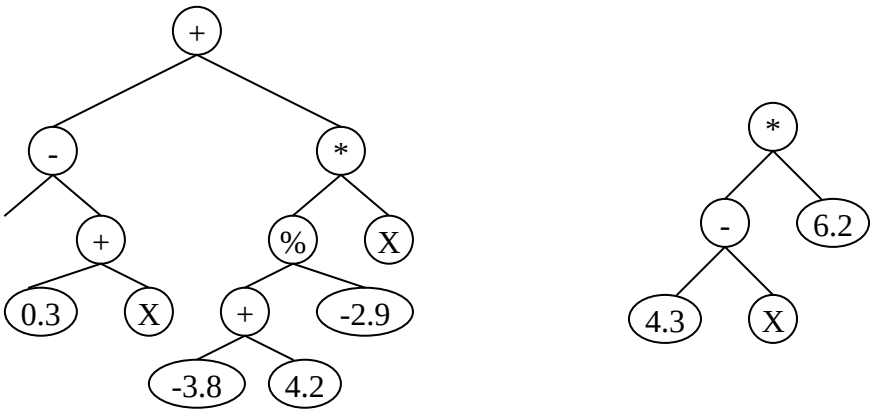


Figura 26. Árbol de mutación y subárbol nuevo generado

Finalmente, para terminar el proceso, se coloca el subárbol en el hueco dejado por el nodo eliminado (Figura 27).

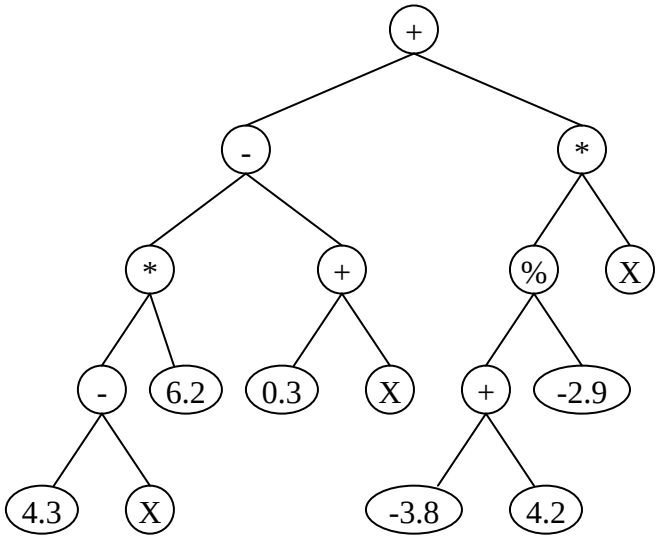


Figura 27. Árbol resultante de la mutación

El operador de mutación provoca en ese individuo un salto en el espacio de estados, comenzando una búsqueda distinta en otra zona. La mayoría de las mutaciones son destructivas, es decir, el individuo empeora, y por eso se utilizan con una probabilidad muy baja, para conseguir variedad genética. Existen estudios sobre la evolución sin el uso de cruces, en los que la mutación juega un papel fundamental (Chellapilla, 1997), en los que se utilizan distintos tipos de mutaciones, pero los resultados siguen siendo peores que utilizando cruces.

2.2.5. Evaluación

Al igual que en los AA.GG., es necesario medir la bondad de cada individuo para realizar la ejecución del proceso evolutivo. De esto se encarga la función de ajuste, que cuantifica la bondad del fenotipo representado por el árbol genético. Existen diversas formas de ajuste, que son las mismas que las explicadas para el caso de AA.GG. y, por lo tanto, son igualmente válidas para PG.

Como se ha explicado anteriormente, uno de los problemas de la PG es el crecimiento excesivo de los árboles para protegerse a sí mismos de los efectos nocivos de los operadores genéticos de cruce y mutación. Para evitar este problema, puede incluirse un factor de parsimonia en el cálculo del ajuste (Soule, 1998; Soule and Foster, 1997). Esta técnica se puede usar para reducir la complejidad del cromosoma que está siendo evaluado, y funciona mediante la penalización en el ajuste del individuo i de la siguiente forma:

$$f(i) = P(i) + \alpha \cdot s_i$$

Donde P(i) es una medida de la bondad del individuo (en este caso, peor cuanto más positivo), α es el nivel de parsimonia y s_i es el tamaño (número de nodos) del individuo. Con este coeficiente se está penalizando el número de nodos de un árbol, y su valor máximo suele ser de 0´1. Con este valor, se necesitará que el árbol tenga 10 nodos para incrementar en una unidad el valor de ajuste. Sin embargo, un valor tan alto ya es muy dañino en la evolución, y se suelen tomar valores menores (0.05, 0.01, etc), dependiendo del rango de valores en los que se espera que estén los ajustes de los individuos.

Dado que es un valor que se suma al valor de ajuste, que se quiere minimizar, constituye una penalización, lo cual provoca que el proceso evolutivo busque árboles sencillos que permitan resolver el problema en cuestión.

2.2.6. Parámetros

Ya que el funcionamiento de la PG es similar al de los AA.GG., los parámetros que controlaban el funcionamiento en éstos también realizan la misma función en la PG. Por lo tanto, los parámetros de tamaño de población, tasa de cruces, probabilidad de mutación, etc., explicados anteriormente, también están presentes aquí. Sin embargo, y dada la distinta naturaleza de la forma de codificación que tiene la PG, que ha obligado a realizar

modificaciones a los operadores genéticos, surge una serie de parámetros especiales de PG, que son los siguientes:

- Altura máxima del árbol. Indica la altura máxima que van a poder alcanzar los árboles durante el proceso evolutivo. Este parámetro se ajusta también de forma proporcional a la complejidad del problema, de forma similar al anterior: un tamaño de población elevado exige elevar la altura máxima, pues, si no, existirá un gran número de individuos repetidos en la población, con predominio de uno de ellos, y los problemas que ello acarrea (fundamentalmente, pérdida de variedad genética y convergencia prematura del algoritmo). Para ajustar este parámetro y el anterior, y valorar la complejidad del problema, es conveniente tener en cuenta el número de elementos que tienen los conjuntos de elementos terminales y no terminales, así como el número de variables y analizar el problema a solucionar.
- La altura máxima que tendrán los árboles iniciales.
- La altura máxima que tendrán los subárboles creados por una mutación de subárbol. Para una de estas mutaciones, el subárbol generado tendrá una altura máxima que será la mínima entre este parámetro y la altura permitida por la restricción de altura máxima en el nodo seleccionado.
- Algoritmos de selección, mutación y creación. Sin embargo, y al contrario de lo que pasaba con los AA.GG., el algoritmo de cruce suele ser fijo y utilizarse siempre el mismo.
- En los cruces, se utiliza también una probabilidad de selección de nodo no terminal, que se usa para escoger nodos del árbol: con esa probabilidad se escogerá un no terminal antes que un terminal (Angeline, 1996; Koza, 1992).
- El coeficiente de "parsimonia" para penalizar árboles grandes.

2.2.7. Aplicaciones

La programación genética es una técnica muy versátil y adaptable, y ha sido utilizada en una gran variedad de campos. Con seguridad su aplicación más utilizada ha sido en el campo de la regresión simbólica: el objetivo es encontrar expresiones que relacionen unas variables de entrada con salidas deseadas. En este campo, la programación genética ha hallado

expresiones que relacionan magnitudes físicas que son tan buenas, o incluso mejores, que las existentes hasta aquel momento en el campo (Babovic et al., 2001a; Babovic et al., 2001b). También ha sido capaz de redescubrir relaciones matemáticas ya existentes, así como relaciones nuevas. Sin embargo, esta técnica no se limita a inducir expresiones matemáticas, sino que su capacidad para construir árboles la convierten en una poderosa técnica de optimización de árboles, y todo lo que pueda ser representado como tales. Algunos campos en los que ha sido aplicado son:

- Predicción en series financieras. Un ejemplo es el artículo publicado por Butler en 1995, en el que se utiliza la programación genética para generar reglas para un sistema experto con el objetivo de realizar apuestas en carreras de caballos (Butler and Tsang, 1995). Un año antes, ya se había publicado otro artículo en el que se había utilizado esta técnica con el mismo fin, apostar en carreras de caballos, con buenos resultados (Perry, 1994).
- Modelización de series en general, no solo financieras (Rivero et al., 2005). Howard Oakley ha utilizado programación genética para modelizar series caóticas basadas en las ecuaciones de Mackey-Glass en datos verdaderos (Oakley, 1994a; Oakley, 1994b).
- Diseño de filtros. En los artículos de Oakley de los años 1993 y 1994 se compara el método de búsqueda con heurística (algoritmo genético convencional) con la programación genética para construir un filtro que elimine el ruido que se obtienen en los datos que se muestrean (Oakley, 1993; Oakley, 1994a). La pila de filtro que se obtuvo usando programación genética apareció como la respuesta apropiada. Actualmente se usa en una aplicación de Sistema Láser.
- Robótica. En este campo las aplicaciones han sido muy variadas. En el trabajo de Spencer de 1994, se describe como ha sido utilizado para generar automáticamente un programa que permita a una criatura de 6 patas andar y arrastrarse, con un operador adicional de perturbación (Spencer, 1994). En el campo de la robótica, la programación genética ha sido muy utilizada en el control de robots, pero también en el comportamiento (por ejemplo, en los trabajos de Reynolds se utilizó para evitar obstáculos (Reynolds, 1992; Reynolds, 1994) y en planificación o incorporación del dominio en el comportamiento de los robots.

- Reconocimiento de patrones y clasificación. Usando patrones del mundo real, se consiguieron clasificadores que formaran propiedades de los datos, con resultados mejores que los obtenidos con otros medios, como la utilización de árboles ID-3, o el uso de un perceptrón multicapa con un algoritmo de aprendizaje de *backpropagation*. Un trabajo teórico es el publicado de Tackett en 1994, en el que se estudia un problema de clasificación utilizando dos estructuras toroidales interconectadas como dos eslabones de una cadena (Tackett and Carmi, 1994). El problema era clasificar una serie de puntos dados en un espacio tridimensional para ver si pertenecen a la estructura. La distribución de los puntos no puede ser linealmente separable por una regla perceptrón, ni pueden ser cubiertos los puntos por conos o hiperconos, ni encerrados por un par de funciones base radiales.
- Redes neuronales. Se ha aplicado la programación genética tanto para el diseño de redes neuronales (Gruau, 1992; Gruau, 1994), como para la creación de nuevos algoritmos de aprendizaje (Segovia and Isasi, 1997).
- Arte. La programación genética se ha aplicado para la generación de imágenes y música de forma semiautomática, con la participación de humanos que evalúen la creación. Sin embargo, en la creación musical la participación humana ha sido posible automatizarla en gran medida (Johanson, 1997; Johanson and Poli, 1998).
- Análisis de bases de datos. Se ha utilizado para distintos objetivos, como por ejemplo construcción de respuestas, clasificación o inducción de reglas, en el campo de la minería de datos (Freitas, 1997).

Estos son solo algunos ejemplos de aplicaciones, pero existen innumerables campos de trabajo abiertos, como son creación de métodos de programación, aplicaciones en el procesamiento del lenguaje natural, en realidad virtual, síntesis de circuitos analógicos y digitales, algoritmos de compresión, procesado de imágenes (Rivero et al., 2004), etc.

II. Estado de la cuestión

Durante los últimos años, la cantidad de datos generados por los estudios de asociación ha aumentado significativamente y, como consecuencia, se han desarrollado numerosos métodos no sólo estadísticos, pero también computacionales, diseñados para realizar análisis de estudios de asociación genotipo-fenotipo basados en SNPs. Estos métodos pueden dividirse en tres grandes grupos (Kelemen et al., 2009): métodos estadísticos, métodos de data mining supervisados y no supervisados, y métodos de soft-computing. Dado que el modelo propuesto en la presente tesis se enmarca en el último grupo, la revisión bibliográfica realizada se centrará principalmente en este grupo.

1. Métodos estadísticos

Existen métodos estadísticos basados en medidas y en tests estadísticos (Balding, 2006; Thomas, 2004). Estos estudios se pueden dividir en análisis preliminar y en técnicas de asociación genotipo-fenotipo. El objetivo del primer grupo es comprobar la calidad de los datos, por ejemplo, comprobando la desviación del Equilibrio Handy-Weinberg (HWE) (Wittke-Thompson et al., 2005). Esto puede ayudar a escoger el conjunto adecuado de SNPs o cómo inferir haplotipos del genotipo. Normalmente, se suele usar la bondad de ajuste de Pearson para analizar la desviación HWE.

Dentro de este grupo estarían también los test para detectar datos perdidos (missing data) (Little and Rubin, 2002). Aunque los datos perdidos no son un problema cuando se analiza un solo SNP, sí lo son cuando se analizan múltiples SNPs, ya que algunos sujetos podrían presentar uno o más genotipos con datos perdidos. La solución más común a este problema es reemplazar estos datos perdidos con valores predichos usando los genotipos de la vecindad. Los métodos más usados para estas predicciones son los métodos de "maximum likehood estimate" o la selección aleatoria basada en una distribución de

probabilidad. La segunda posibilidad tiene como ventaja sobre la primera que las repeticiones de selecciones aleatorias suavizan los efectos causados por la pérdida de datos en los resultados.

Existen principalmente tres posibilidades a la hora de realizar análisis de asociación. La primera corresponde con las pruebas de asociación de un solo SNP. Este tipo de test se basa en probar la hipótesis nula de que no existe ninguna asociación entre los valores de cada SNP y el valor caso/control. Para probar esto, los test más usados son el test de Pearson y el test de Fisher, especialmente este último a pesar de que es computacionalmente más costoso. En los estudios de asociación complejos la asociación entre el desarrollo de una enfermedad y el valor de un solo SNP es muy débil, por lo que el test de Fisher y el test de Pearson pueden no ser suficientemente potentes. En estos casos, se puede aplicar el test de Cochran-Armitage (Armitage, 1995) que es más conservador que los anteriores y no depende de del HWE.

El segundo tipo de análisis se refiere a los estudios de asociación con salidas continuas. Los métodos clásicos para estos estudios son la regresión lineal o el análisis de varianza (ANOVA). ANOVA es equivalente al test de Pearson y compara la hipótesis nula de no-asociación con la alternativa general. La regresión lineal reduce los grados de libertad asumiendo que existe una relación lineal entre el valor de salida considerado en el estudio y el genotipo. En ambos casos el test necesita que los valores del genotipo sigan una distribución normal para cada genotipo. Si los estudios son del tipo caso/control, la regresión lineal no es aplicable debido a la distribución no-lineal de los datos del fenotipo: caso o control. La regresión logística es una aproximación más elaborada, ya que suele ser usada para asignar una puntuación para una predicción en lugar de proporcionar una predicción en sí. Además de esto, la regresión logística es un método muy flexible que puede ser fácilmente adaptado para múltiples SNPs y permite incluir interacciones ambientales así como covariantes como el sexo o la edad. Cuando los test se aplican a múltiples SNPs, el objetivo del test es, dado un conjunto de SNPs en sujetos caso y control, encontrar un conjunto de estos relacionado con la enfermedad, y/o cuando la asociación sea dada, encontrar el SNP más cercano al polimorfismo casual. Cuando el número de SNPs es demasiado grande, la capacidad de seleccionar los SNPs más relevantes y borrar aquellos que presenten correlaciones grandes con otros aumenta el poder del análisis, a expensas de perder algo de información.

Otra solución al problema de tener demasiados SNPs correlacionados es usar técnicas como "stepwise selection procedure" (Cordell and Clayton, 2002) o "bayesian shrinkage methods" (Wang et al., 2005). Uno de los mayores problemas del análisis de múltiples SNPs está relacionado con la existencia de demasiados predictores, algunos altamente relacionados. Una de las estrategias inspirada en la estructura de bloques seguida por el genoma humano, es el uso de haplotipos para eliminar las correlaciones en las regiones de baja recombinación. Esta aproximación permite obtener análisis con un menor número de grados de libertad e incluso subraya la importancia de los efectos combinados de estas variables accidentales.

Finalmente, Briggs et al. estudiaron interacciones genéticas (epistasis) siguiendo un enfoque estadístico, combinando varios métodos analíticos (Briggs et al., 2010). Utilizaron un análisis multi-fase que combinaba técnicas de aprendizaje máquina supervisadas con la regresión logística, aplicado a una enfermedad autoinmune: la artritis reumatoide.

2. Métodos de data mining

Los métodos de data mining tienen un numero de características que los hacen muy atractivos para realizar análisis de estudios de asociación. Estos métodos son normalmente computacionalmente eficientes y altamente escalables cuando se trata con una alta dimensionalidad de los datos, como ocurre en los estudios de asociación. Los modelos obtenidos a partir de estas técnicas tienden a ser considerados más simples en comparación con aquellos obtenidos a partir de técnicas estadísticas, a pesar de que estos métodos permiten la explotación y descubrimiento, es decir, que al contrario de los métodos estadísticos que son dirigidos a hipótesis, los métodos de data mining son generadores de hipótesis. Los resultados obtenidos por análisis que usan estas técnicas son a menudo considerados como un complemento a los análisis estadísticos clásicos. Finalmente, las técnicas de data mining normalmente usan datos discretos y manejan estructuras simbólicas, proporcionando explicaciones y resultados más útiles y fáciles de entender que los proporcionados por los métodos estadísticos, que normalmente son más complejos.

A menudo se define pobremente data mining como "la extracción no trivial de información implícita, previamente desconocida y potencialmente útil de los datos". Las técnicas de data mining aplicadas al análisis de estudios de asociación se pueden categorizar en dos grupos: métodos de clasificación que intentan encontrar marcadores y otras

características relevantes para predecir una enfermedad y las técnicas de clustering que intentan encontrar subgrupos de sujetos basándose en su similitud genotípica y fenotípica.

2.1. Métodos de clasificación

Los métodos de clasificación intentan encontrar reglas o patrones que ayuden a predecir el valor de una variable en función de variables independientes. Esto aplicado a estudios de asociación implica encontrar una serie de patrones de SNPs o de haplotipos que juntos formen un buen predictor de un fenotipo y que puedan predecir la susceptibilidad a una enfermedad del individuo. Una de las técnicas más populares son los denominados árboles de regresión/clasificación/decisión. Estas técnicas están basadas en la partición recursiva de los datos, resultando un modelo en forma de árbol. Ejemplos de estas técnicas aplicadas al análisis de estudios de asociación pueden encontrarse en diversas referencias (Cook et al., 2004; Uhmn et al., 2009; Young and Ge, 2005).

El Symbolic Discriminant Analysis (SDA) se utiliza en el análisis de grandes conjuntos de datos y realiza automática y simultáneamente una reducción de variables y el desarrollo del modelo. Se pueden encontrar algunos ejemplos de la aplicación de estas técnicas al campo de la genómica en la literatura (Reif et al., 2004). En otro trabajo publicado también en 2004 se muestra un ejemplo de asociación utilizando reglas de asociación (Rova et al., 2004). Este método se basa en la generación de reglas que describen co-ocurrencias de conjuntos de características. Otros métodos como DICE (Tahri-Daizadeh et al., 2003), la reducción de dimensión multifactorial (Moore, 2004; Ritchie et al., 2001) o las máquinas de soporte vectorial (SVM) (Ban et al., 2010; Waddell et al., 2005) han sido ampliamente utilizados también para el análisis de estudios de asociación.

2.2. Métodos de clustering

Los métodos de clustering intentan localizar subgrupos relativamente homogéneos dentro de un conjunto de datos. En el contexto de los estudios de asociación, los algoritmos de clustering intentan encontrar subgrupos de individuos que potencialmente compartan características genéticas. Este clustering puede aplicarse tanto en genotipo como en fenotipo. Una vez aplicado el clustering, debería ser más sencillo localizar factores

genéticos dentro de cada cluster, aumentando la potencia estadística, aunque reduciendo el número de patrones. La aplicación de estas técnicas puede encontrarse en la literatura (Molitor et al., 2003; Toivonen et al., 2005; Tzeng et al., 2003; Wilcox et al., 2003).

3. Métodos de soft-computing

Las técnicas de soft-computing abarcan tres de las principales tecnologías orientadas al desarrollo de sistemas inteligentes, llamadas lógica difusa, redes de neuronas y sistemas bioinspirados como la computación evolutiva, sistema inmune artificial o swarm intelligence. Dentro de esta familia de técnicas, existen dos tareas principales en lo que se refiere al análisis de estudios de asociación.

La primera tarea es el descubrimiento de patrones de SNPs para enfermedades complejas, que incluye la selección de SNPs útiles de entre miles de SNPs asociados con una enfermedad. Existen dos aproximaciones principalmente: en primer lugar la búsqueda de tagSNPs, que consiste en la búsqueda no supervisada de bloques de SNPs relacionados con la enfermedad; la otra aproximación consiste en la búsqueda de patrones de interacción SNP-enfermedad, basada en métodos supervisados, que relacionan conjuntos de SNPs con enfermedades, sin tener en cuenta bloques preestablecidos.

La segunda tarea se ocupa del modelado de interacciones gen-gen y gen-ambiente en enfermedades complejas. Debido a la complejidad del problema existen relativamente pocos trabajos que puedan abordar este problema.

Jourdan et al. utilizan un algoritmo genético para inferir reglas a partir datos de SNPs procedentes de estudios de ligamiento para asociar SNPs con diabetes (Jourdan et al., 2002). Posteriormente, mejoraron el algoritmo para aplicarlo en entornos distribuidos (Vermeulen-Jourdan et al., 2005).

Clark et al. desarrollaron un algoritmo genético que permite construir árboles lógicos (Clark et al., 2008; Clark et al., 2005). Estos árboles lógicos consisten en expresiones lógicas (nodos) que agrupan conjuntos de SNPs (hojas). Este algoritmo genético fue aplicado a estudios sobre la variación genética de genes candidatos. En cada generación del algoritmo genético se modifica la población por medio de las técnicas de mutación, selección y cruce. Los mejores árboles se seleccionan basándose en la función de fitness, que consiste en la salida de la regresión bayesiana de cada árbol.

Estudios recientes muestran que la programación genética mejora el rendimiento de muchas técnicas estadísticas tradicionales, así como de muchos métodos de data mining o machine learning, como la regresión lineal o las SVM. Ritchie et al. usan una red de neuronas cuya arquitectura ha sido optimizada utilizando programación genética para identificar una combinación de genes relacionados con el riesgo de desarrollo de una enfermedad (Ritchie et al., 2003). Motsinger et al. también usan una red de neuronas optimizada usando programación genética para detectar interacciones gen-gen en datos de SNPs (Motsinger et al., 2006).

Hubley et al. presentan un algoritmo evolutivo para selección de SNPs multiobjetivo, que es capaz de aproximar un conjunto de soluciones óptimas. Este diseño funciona muy bien con grandes estudios (Hubley et al., 2003). La implementación consiste en una versión modificada del algoritmo evolutivo de Strength-Pareto. Este algoritmo es especialmente adecuado para resolver problemas de optimización con varios objetivos o en espacios de búsqueda complejos donde heurísticas más exhaustivas no pueden ser usadas.

Banzhaf et al. usan algoritmos genéticos así como otras técnicas de computación evolutiva para modelar relaciones gen-gen, usando árboles evolucionados para realizar el análisis de haplotipos (Banzhaf et al., 2006).

Moore y White desarrollaron un método hibrido que combina la programación genética con la reducción de dimensionalidad multifactorial (MDR) para escoger SNPs (Moore and White, 2006). Este método funciona mejor que los métodos de Random Search cuando se usan datos simulados.

Hasta el momento se han repasado distintas aproximaciones para realizar el análisis de los estudios de asociación. A continuación se hará un estado de arte sobre el uso de la técnica empleada en este trabajo para inferir información de los datos de SNPs, denominada reglas de asociación (Hernández et al., 2004; Tan et al., 2006).

Las reglas de asociación son una estructura muy popular para expresar patrones en un conjunto de datos. Estos patrones pueden ser útiles para entender el comportamiento general del problema que genera el conjunto de datos y de esta manera se tenga más información que pueda asistir a problemas de toma de decisiones, diagnóstico, etc. El conjunto de datos puede estar expresado en forma de tabla, donde las filas se interpretan

como ejemplos mientras que las columnas son los tipos de características que presenta cada ejemplo.

Una regla de asociación es una proposición probabilística sobre la ocurrencia de ciertos estados en los datos. Una típica regla de asociación podría ser "SI C1=0 Y C2=4 ENTONCES CONSECUENTE". Formalmente, una regla de asociación puede verse como reglas de la forma "SI A ENTONCES B", donde A y B son conjuntos de ítems disjuntos. Dada una regla de asociación, se suele trabajar con dos medidas de calidad: la cobertura y la confianza. La cobertura de una regla se define como el número de instancias que la regla predice correctamente. Por otra parte, la confianza mide el porcentaje de veces que la regla se cumple cuando se puede aplicar. Los distintos algoritmos de búsqueda de reglas de asociación se basan en la búsqueda de reglas que cumplan unos requisitos mínimos en cuanto a cobertura y confianza. Diversas implementaciones de reglas de asociación pueden encontrarse en los siguientes trabajos (Deshpande and Karypis, 2002; Hipp et al., 2000; Li et al., 2001; Liu et al., 1998; Liu et al., 2001; Lucrédio et al., 2004; Srikant and Agrawal, 1995).

3.1. Utilización de los algoritmos evolutivos para extracción de reglas

Los algoritmos evolutivos tienen un carácter de búsqueda global que hace que sean especialmente adecuados para resolver problemas presentes en las distintas etapas del proceso de descubrimiento de conocimiento. En procesos de extracción de reglas, los algoritmos evolutivos tratan de forma adecuada las interacciones entre atributos ya que evalúan una regla como un todo mediante la función de ajuste, en lugar de evaluar el impacto de añadir o eliminar una condición de una regla, como ocurre en los procesos de búsqueda local incluidos en la mayoría de los algoritmos de inducción de reglas y árboles de decisión.

Entre las distintas clases de algoritmos evolutivos, los algoritmos genéticos y la programación genética son los más utilizados para el descubrimiento de reglas. Estas dos clases de algoritmos difieren fundamentalmente en la representación de los individuos. En el caso de los algoritmos genéticos, los individuos se representan como una cadena lineal de condiciones, donde cada condición suele ser una pareja atributo-valor, mientras que en

la programación genética un individuo suele representarse mediante un árbol donde los nodos hoja son los valores de atributos y los nodos internos representan las funciones.

3.1.1. Utilización de AA.GG.

Los algoritmos genéticos siguen dos enfoques respecto a la forma de codificar reglas dentro de una población de individuos (De Jong, 1988): por un lado el enfoque "cromosoma-regla", donde cada individuo codifica una sola regla, y por otra parte el enfoque "cromosoma-base de reglas", también denominado enfoque Pittsburg (Smith, 1980), donde cada individuo representa un conjunto de reglas.

Dentro del enfoque "cromosoma-regla", existen también dos propuestas. La primera, denominada enfoque Michigan (Holland and Reitman, 1978), en la que cada individuo modifica una sola regla pero la solución final será la población final o un subconjunto de la misma. En este caso es necesario evaluar el comportamiento del conjunto de reglas al completo y la aportación de la regla individual al mismo. Por otro lado el enfoque IRL (*Iterative Rule Learning*) (Venturini, 1993), en la que cada cromosoma representa una regla y la solución particular es el mejor individuo del algoritmo genético, mientras que la solución global consiste en el conjunto de los mejores individuos obtenidos a partir de varias ejecuciones del algoritmo genético.

Escoger una aproximación u otra depende de la tarea que el algoritmo de búsqueda de reglas realice. Si el objetivo es obtener un conjunto de reglas de clasificación, entonces deberá ser evaluado por el comportamiento del conjunto total de reglas y no solo por la calidad de una regla simple. En este caso, la mejor aproximación sería seguir la solución "cromosoma = base de reglas", que considera la interacción entre reglas. Se pueden encontrar algunos ejemplos en los que los algoritmos genéticos son usados para la obtención de reglas de asociación y usan la anterior aproximación (De Jong et al., 1993; Janikow, 1993). Es necesario tener en cuenta que esta aproximación presenta algunos problemas relacionados con el uso de grandes individuos (en los casos de cromosomas de longitud variable), incrementando el coste computacional del algoritmo. Debido a esto también han sido desarrolladas aproximaciones siguiendo el modelo "cromosoma = regla" (Giordana and Neri, 1995; Greene and F., 1993). En estos ejemplos se utilizan unos individuos más reducidos, simplificando el diseño de los operadores genéticos. Este enfoque tiene dos inconvenientes, por un lado, la dificultad del cálculo de la función de

fitness, ya que cada regla se evalúa individualmente y es difícil determinar la calidad del conjunto de reglas al completo. Por otra parte, puesto que el objetivo es obtener un conjunto de reglas, el algoritmo no debería converger hacia un único individuo. Para evitar esto se necesita alguna técnica de nichos (Beasly et al., 1993) que fomente la existencia de distintos individuos dentro de la población.

En procesos de descubrimiento de reglas de asociación, es más adecuado el enfoque "cromosoma= regla", ya que el objetivo suele ser encontrar un conjunto de reglas en las que la calidad de la regla se evalúa de forma independiente al resto. A continuación se describen varios trabajos que combinan AA.GG. con el enfoque IRL.

GENAR (Mata et al., 2001) es una herramienta que fue diseñada para el descubrimiento de reglas de asociación en bases de datos que contenían atributos cuantitativos. Los autores utilizan un algoritmo evolutivo para obtener diferentes intervalos, así como IRL para evitar obtener siempre la misma regla.

Hoffman presenta un algoritmo de *boosting* basado en el enfoque IRL para el diseño de un sistema de bases de reglas difusas (Hoffmann, 2004). La base de reglas difusas se generó utilizando un algoritmo evolutivo que optimizaba un clasificador de reglas difusas de cada vez. Esta técnica reduce, de forma sistemática, los pesos de los ejemplos correctamente clasificados para centrarse, en las siguientes iteraciones del método de generación de reglas, en aquellos ejemplos difíciles de aprender.

ARMNGA (Association Rules Mining in Novel Genetic Algorithm) (Dai et al., 2007) es un algoritmo de minería espacial que utilizó un AG diseñado específicamente para el descubrimiento de reglas de asociación. Yan et al., por otra parte, diseñaron una estrategia basada en AA.GG. para la identificación de reglas de asociación sin la necesidad de especificar un umbral de cobertura mínima (Yan et al., 2009).

Qodaman et al. también propusieron un método basado en AA.GG. (Qodmanan et al., 2011). Este método extraía las reglas que tenían la correlación más alta entre la cobertura y la confianza. Al igual que una de las aproximaciones descritas anteriormente, no era necesario especificar un umbral de cobertura mínima, pero tampoco una confianza mínima.

Yang et al. presentaron un método basado en AA.GG. que evaluaba el efecto combinado de 26 SNPs involucrados en rutas metabólicas relacionadas con el cáncer de mama (Yang et al., 2011).

Li et al. combinaron un AG con *Linear Discriminant Analysis* (LDA) para desarrollar un método que tratase de diferenciar cáncer de nasofaringe de tejido normal de acuerdo a variables que pertenecían a regiones seleccionadas relacionadas con proteínas, ácidos nucleicos y lípidos (Li et al., 2012).

Finalmente, Mooney et al. aplicaron AA.GG. a la búsqueda de asociaciones multi-locus (Mooney et al., 2012). Para ello, exploraron y evaluaron el uso de un AG para el descubrimiento de grupos de 2, 3 ó 4 SNPs aplicado a la detección de asociaciones con el trastorno bipolar.

3.1.2. Utilización de PG

En relación a la PG, se pueden encontrar diversos estudios en los que esta técnica ha sido aplicada a enfermedades concretas. Un ejemplo es el estudio llevado a cabo en Curitiba (Brasil) en el que se aplicó esta técnica en la investigación sobre el dolor de pecho (Bojarczuk et al., 2000). El objetivo del estudio era encontrar reglas de clasificación que permitiesen diagnosticar diversas enfermedades que tenían como síntoma el dolor de pecho. Este estudio incluía 165 variables que se estimaban relevantes para clasificar hasta 12 enfermedades distintas. A pesar de obtener resultados aceptables, los pocos casos con los que se trabajaba hicieron imposible obtener mejores conclusiones. Sin embargo, este estudio ya mostraba lo prometedor de esta técnica y la importancia de la selección de las variables consideradas en el estudio.

En 2001 aparecieron dos nuevos estudios que avanzaban en el uso de la PG con fines médicos. Uno de ellos proponía un modelo basado en árboles para predecir casos de cáncer de mama (Kuo et al., 2001). El objetivo de dicho estudio era desarrollar un sistema de clasificación previo al análisis del caso por un experto y obtuvo resultados tan satisfactorios que superaron el 85% de acierto. En el segundo estudio (Brameier and Banzhaf, 2001) se realizó una comparación entre diversas técnicas basadas en PG con el objetivo de encontrar la técnica que ofrecía mejores resultados con menor coste computacional.

En 2004 se publicaron diversos estudios que buscaban, entre otros, predecir la supervivencia de pacientes con cáncer (Langdon and Buxton, 2004) en base a su ADN y al

desarrollo de la enfermedad, o clasificar el ADN para detectar la posible presencia de linfomas (Hong and Cho, 2004), ambos alcanzando altísimos índices de acierto.

Fue también en este mismo año cuando terminaron de unirse los conceptos de PG, el descubrimiento de reglas de clasificación y su uso en bases de datos médicas, tal y como defendieron estudios de la Universidad de Kent (Bojarczuk et al., 2004), donde se analizó la necesidad de buscar una sintaxis y unas restricciones que hicieran que las reglas fueran más simples y fáciles de comprender. Para este estudio se trabajó sobre varias enfermedades distintas (algunas ya vistas, como el cáncer de mama o el dolor de pecho y otras nuevas, como algunas relacionadas con afecciones dermatológicas), comparando diferentes métodos de clasificación con el objetivo de mejorar los resultados.

La proliferación de estos trabajos y los buenos resultados que obtenían invitaron a que se siguiera investigando en de qué forma se podía hacer más eficiente la PG a la hora de detectar enfermedades o la predisposición a ellas, destacando, entre otros, un estudio sobre tres técnicas para encontrar un patrón común en el cáncer de mama (Delen et al., 2005) realizado en 2005. También es reseñable un estudio que pretendía buscar nuevas formas de afrontar el problema, probando, por ejemplo, el uso de gramáticas que modelasen el ADN (Langdon et al., 2010) de manera similar a como se llevará a cabo en esta tesis.

En 2006, se llevó a cabo un estudio en la Universidad de Liverpool (Guo and Nandi, 2006) en el que se planteaba de nuevo este método para tratar de clasificar los datos obtenidos mediante técnicas estadísticas, como el análisis discriminante de Fisher, combinadas con clasificadores más simples, como el de mínima distancia. Tratando también la detección del cáncer (esta vez de vesícula) y utilizando validación cruzada (de manera similar a cómo se realiza en esta tesis), en otro estudio de 2006 se obtuvieron resultados de hasta el 81% de acierto, además de lograr aislar de manera satisfactoria ciertos genes que se dedujeron relevantes (Mitra et al., 2006).

En 2009, en el campo de la dermatología, se llevó a cabo un estudio aplicado a casos de melanomas. El objetivo de dicho estudio era refinar parámetros de algoritmos de clasificación utilizando Programación Genética (Winkler et al., 2009). Entre el 2009 y el 2010 también se presentaron varios estudios enfocados a obtener modelos de predicción de desarrollo de la enfermedad (Paul and Iba, 2009) o del índice de supervivencia de aquellos individuos que la hayan padecido (Farinaccio et al., 2010).

Finalmente, en los últimos años ha aumentado el número de estudios que aplican la Programación Genética para resolver problemas de predicción en el ámbito médico, como el llevado a cabo por Engoren et al. para la predicción de la aceptación del individuo tras una operación de corazón (Engoren et al., 2013), o el trabajo desarrollado por Canavan et al. que tiene como objetivo monitorizar mediante Programación Genética la evolución de una de las afecciones más recurrentes en el cáncer de mama (Canavan et al., 2012).

4. Consideraciones

Tras la revisión de los distintos estudios existentes en relación con el trabajo presentado se han obtenido varias conclusiones.

En lo relativo a estudios basados en técnicas estadísticas, se ha observado que, en ciertos casos, no son lo suficientemente potentes para detectar riesgos relativamente bajos y asociaciones obtenidas previamente en experimentos biológicos. Además, en ciertos estudios se requieren datos cuya obtención resulta en un coste prohibitivo y, por lo tanto, su aplicación se ve reducida. En otros casos, se observa, también, una dependencia entre el buen funcionamiento de los métodos propuestos y condiciones ideales que no se dan en situaciones reales, provocando una disminución de la precisión del método aplicado. Finalmente, este tipo de técnicas no permiten la variabilidad y aleatoriedad que sí permiten las técnicas basadas en inteligencia artificial.

En cuanto a los estudios basados en técnicas de inteligencia computacional, los que utilizan un cierto diseño de experimentos, muchas veces, asumen que se posee cierta información que, en general, no se tiene debido al alto coste mencionado anteriormente y, por ello, las tecnologías de genotipado existente no la suelen proporcionar. Ciertos métodos, además, poseen restricciones computacionales, como no ser capaz de procesar poblaciones de más de mil sujetos o no ser capaz de inferir ciertos parámetros importantes que deberán ser introducidos por el usuario, lo cual puede redundar en un mal funcionamiento del sistema.

En relación a los métodos basados en PG, uno de los problemas que presentan es que las soluciones obtenidas pueden ser demasiado complejas, dificultando su comprensión y evaluación, haciendo que la elaboración de conclusiones se complique. En los trabajos descritos en el apartado anterior, también se ha visto que otro de los aspectos que suele quedar descuidado es el proporcionar más información sobre las soluciones, sea estadística

o mediante un análisis posterior de las propias soluciones, que permita sacar conclusiones sin un esfuerzo grande por parte del usuario. Estos factores hacen que pese a que la PG pueda ser una técnica apropiada, sin el apoyo de más información o intentar reducir la complejidad de la solución, en ocasiones sean menos útiles los resultados que aporta dada la dificultad de su interpretación.

III. Hipótesis

En la sección inmediatamente anterior se han presentado una serie de técnicas y métodos que han sido aplicados al análisis de estudios de asociación. En el prólogo, además, se ha visto que la minería de reglas de asociación puede ser modelada como un problema de optimización combinatoria. En base a esto, en esta tesis se propone la siguiente hipótesis: "Es posible desarrollar un método basado en Computación Evolutiva que sea capaz de realizar minería de reglas de asociación modelado como un problemas de optimización combinatoria".

Adicionalmente, en la aplicación biomédica concreta que se propone en esta tesis, se plantea la siguiente hipótesis: "La predisposición genética está subyacente en los datos genéticos en algún tipo de estructuras y las técnicas de Computación Evolutiva son adecuadas para extraer estas estructuras".

En trabajos previos realizados por la autora (Aguiar-Pulido et al., 2010; Aguiar et al., 2009) se presentó una versión inicial de la primera aproximación incluida en esta tesis, basada en AA.GG., en los que ya se dejaba entrever el potencial que este modelo ofrecía. En uno de estos trabajos se validaba esta versión inicial del modelo propuesto sobre las bases de datos del UCI *Machine Learning Repository*, utilizadas como referencia a la hora de validar técnicas de aprendizaje máquina.

A diferencia de los AA.GG., la PG ofrece una mayor flexibilidad gracias a la representación utilizada para obtener las expresiones que conformarán posibles soluciones al problema. Por ello, y como corolario, se propone una segunda aproximación en el desarrollo del modelo propuesto, utilizando como núcleo un algoritmo de PG, que, además, realice un análisis de las expresiones obtenidas para facilitar al usuario la obtención de conclusiones. Las dos aproximaciones serán comparadas con el objetivo de determinar qué tipo de estructura es más adecuada para la representación de la predisposición genética, y si la

flexibilidad que aporta la PG a la hora de representar las expresiones extraídas redunda en una mejora de los resultados del modelo.

IV. Modelo propuesto

En este trabajo se propone el uso de CE para realizar minería de reglas de asociación, modelado como un problema de optimización combinatoria. Se plantean dos aproximaciones que se diferencian en el núcleo: una está basada en AA.GG. y otra en PG. A continuación se describe la estructura global y los distintos procesos que componen el método desarrollado.

1. Estructura global

En primer lugar, se divide el conjunto original de datos en dos partes: la primera parte, que de ahora en adelante se referenciará como conjunto de entrenamiento, se utilizará para la fase de entrenamiento y la segunda parte, denominada conjunto de test, se utilizará para validar el algoritmo propuesto. Tras dividir el total de los datos, el conjunto de entrenamiento se utilizará como entrada a lo que de ahora en adelante se denominará proceso iterativo. Este proceso, que se explica más detalladamente a continuación, se ejecutará un máximo de diez veces. En cada ejecución, se devuelve una expresión candidata que podrá ser almacenada en el conjunto final de expresiones, o descartada si no supera unos ciertos umbrales de calidad fijados previamente. Finalmente, una vez que se han realizado todas las ejecuciones del proceso iterativo, se clasificarán los datos del conjunto de test utilizando las expresiones almacenadas en el conjunto final de expresiones, obteniendo la información pertinente relativa al funcionamiento del modelo sobre dichos datos.

Para mayor robustez, se ha utilizado la técnica de validación cruzada. La validación cruzada, o cross-validation, es una técnica empleada para estimar el rendimiento de un modelo predictivo. Por ello, y con el objetivo de garantizar que las medidas obtenidas (por ejemplo, porcentajes de clasificación) son independientes de los datos utilizados para entrenar y

validar el método, se divide el conjunto total de datos utilizando un porcentaje para entrenar el modelo y el restante para validarlo, repitiendo el proceso un número determinado de veces. Por ejemplo, si se considera una validación cruzada de 10-fold, como la utilizada en esta tesis, el 90% de los datos se utilizarían para entrenar el método desarrollado, mientras que el 10% restante se utilizaría durante la fase de test, repitiendo este proceso diez veces. De este modo, a lo largo de las ejecuciones, todos los datos serían utilizados para entrenar y para validar el método.

La estructura global del método propuesto se muestra en la Figura 28.

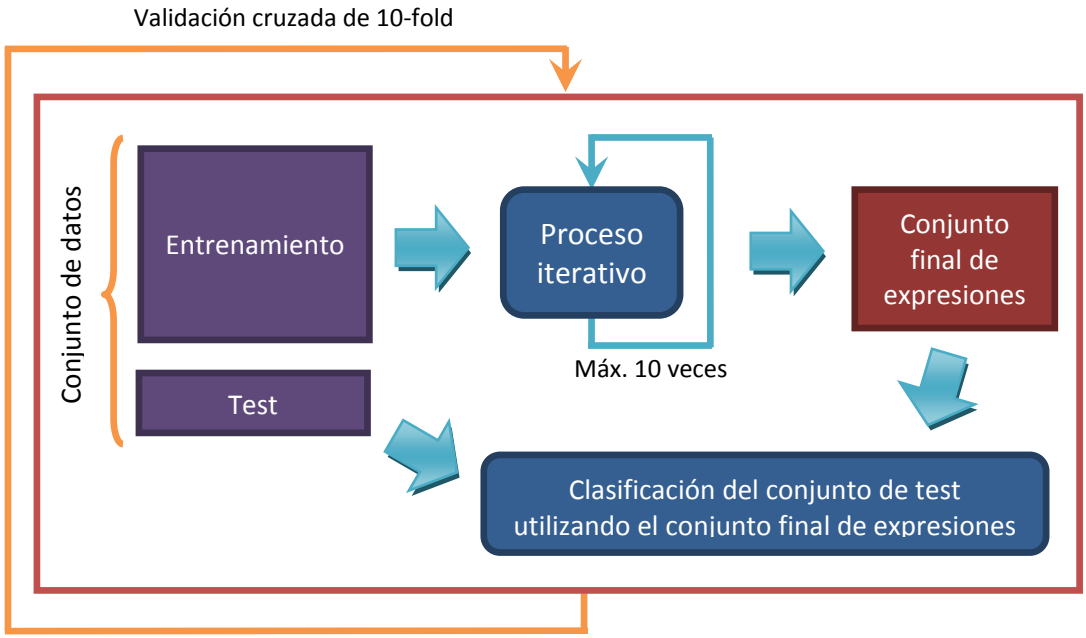


Figura 28. Estructura global del método

2. Proceso iterativo

A continuación se describe el funcionamiento del proceso iterativo. Este proceso representa el núcleo del método desarrollado y utiliza como base un algoritmo evolutivo que, en función de la aproximación escogida, puede ser un AG o PG. En cualquier caso, cada individuo del algoritmo evolutivo corresponde a una expresión, de forma similar a cómo funciona el enfoque IRL presentado en el "Estado de la cuestión". Como se puede ver en la Figura 29, cada iteración de este proceso incluye los siguientes pasos:

1. En primer lugar, se inicializa la población del algoritmo evolutivo de forma aleatoria y se realiza el cálculo inicial de los valores de fitness o ajuste.

2. A continuación, se evoluciona el algoritmo evolutivo utilizado como núcleo hasta que se cumple alguna de las condiciones de parada (parámetros configurables por el usuario):
    i. El fitness es menor que un cierto umbral.
    ii. Se ha alcanzado un número máximo de iteraciones.
3. Llegados a este punto, se ordena la población en base a los valores de fitness, de tal forma que la primera posición sea ocupada por el mejor individuo.
4. Una vez que ha finalizado el proceso evolutivo, se escoge el mejor individuo. Este individuo, que representa una expresión, deberá clasificar un porcentaje mínimo (fijado por el usuario) de los datos para que sea añadido al conjunto final de expresiones, en caso contrario será descartado. Por lo tanto, si dicho individuo cumple los criterios de calidad establecidos, se marcan los datos que cubre la expresión y que han sido correctamente clasificados para que no se tengan en cuenta a la hora de entrenar el algoritmo evolutivo en la siguiente iteración del proceso iterativo.
5. Si la expresión es descartada y no se ha superado el número máximo de iteraciones (10 iteraciones), se vuelve a iniciar el proceso. Si, por el contrario, dicha expresión supera el umbral de calidad establecido, se comprueba si todos los datos han sido clasificados, en cuyo caso finalizaría la ejecución del proceso iterativo. Si esto no es así, y no se ha superado el número máximo de iteraciones anteriormente mencionado, se volvería a iniciar el proceso.

Los datos utilizados como entrada (en este caso aquellos que pertenecen al conjunto de entrenamiento) se marcan como clasificados para que el método sea capaz de encontrar, además de expresiones generales, expresiones más específicas o que afecten a un porcentaje más bajo del conjunto de datos. De esta forma, los datos marcados no serán tenidos en cuenta, como entrada, en la siguiente iteración del proceso.

El proceso descrito se ejecutará, por lo tanto, un máximo de diez veces, a no ser que todos los datos utilizados como entrada estén marcados como clasificados, en cuyo caso la ejecución de este proceso finalizaría y se pasaría al siguiente bloque de la estructura global: la clasificación de los datos pertenecientes al conjunto de test.

Finalmente, es necesario destacar que en el caso de la aproximación 1, basada en AA.GG., las expresiones se corresponderán con reglas, mientras que en el caso de la aproximación 2, basada en PG, estas se corresponderán con árboles.

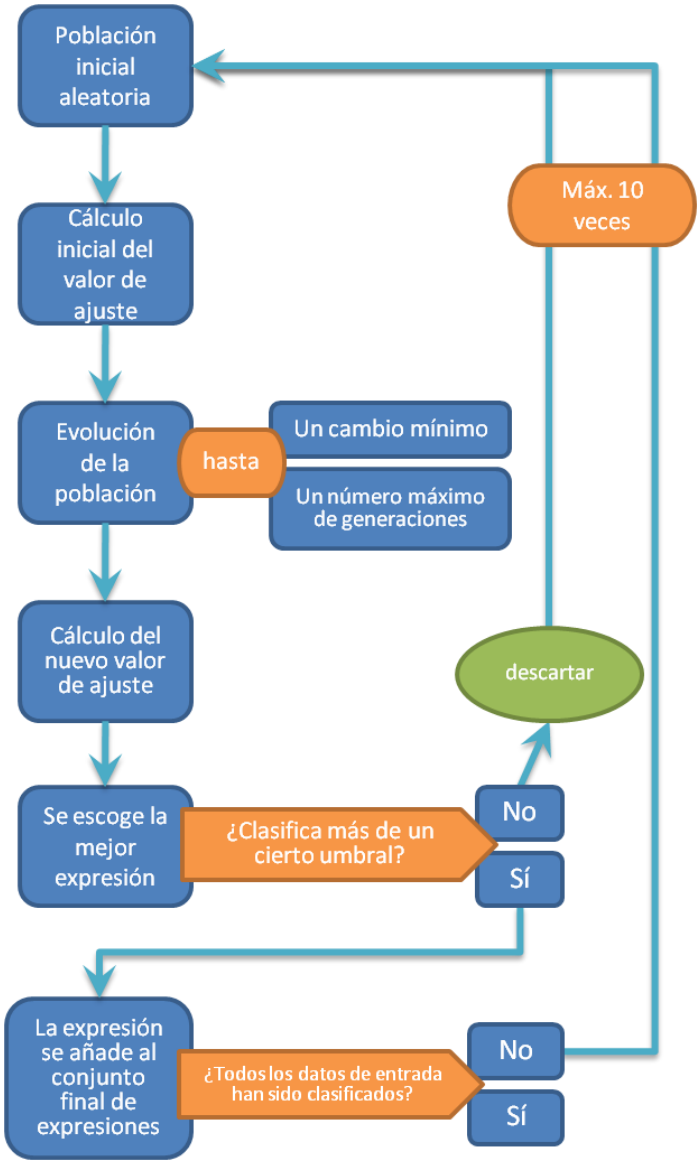


Figura 29. Proceso iterativo

2.1. Algoritmo evolutivo

Como ya se ha mencionado previamente, el núcleo de la solución propuesta en este trabajo de tesis utiliza como base un algoritmo evolutivo. Se han desarrollado dos aproximaciones: una basada en AA.GG. y otra en PG.

2.1.1. Aproximación 1: núcleo basado en AA.GG.

Codificación

Cada individuo de la población representa una expresión candidata. En este caso, al tratarse de un AG, dicha expresión se corresponderá con una regla, siguiendo por lo tanto el enfoque IRL. La estructura de un individuo del algoritmo genético estará compuesto por 2n+1 genes que forman un cromosoma, donde n es el número de variables independientes o de entrada. Este cromosoma representa una posible regla que incluye varios operadores AND. En cada cromosoma, se codifica si las variables están activas o no para la regla, es decir, si van a formar parte de ella, y su valor. Por lo tanto, cada individuo del AG será codificado como una lista de números, en donde cada posición o gen puede tomar valores enteros. Dada la naturaleza del problema a resolver, sólo se permite una salida binaria. Por lo tanto, las reglas candidatas obtenidas por esta aproximación tendrán el siguiente aspecto:

$$IF\ variable_1 = valor_1\ AND\ \dots\ AND\ variable_n = valor_n\ THEN\ salida_i$$

En donde $salida_i$ puede tomar el valor 0 ó 1. De este modo, si se considera el problema al que pretende aplicarse el método propuesto, podría obtenerse un individuo como el mostrado a continuación.

-1	-1	-1	2	-1	-1	1	1

Figura 30. Posible regla candidata

En este caso, cada uno de los valores que pueden tomar los genes que forman el cromosoma (-1, 0, 1 ó 2) tendrían un sentido biológico, excepto el -1 que representa que esa variable no forma parte de la regla (no está activa). Por lo tanto, la codificación del individuo mostrada en dicha figura representa la regla:

$$IF\ SNP_4 = 2\ AND\ SNP_7 = 1\ THEN\ predispuesto$$

Función de fitness

La función de fitness fue diseñada de tal forma que se tuviesen en cuenta una serie de factores que, con diferentes pesos establecidos de forma empírica, penalizasen características no deseables, como los falsos positivos y falsos negativos o las reglas demasiado largas. El fitness es calculado siguiendo el algoritmo detallado a continuación:

1. Evaluación de la expresión sobre el conjunto de datos de entrenamiento.

2. Cálculo del número de variables activas, es decir, aquellas que forman parte de la regla candidata.
3. Cálculo del número de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN).
4. Cálculo del número de aciertos en base a la siguiente fórmula:

$$Aciertos = VP + VN$$

5. Cálculo del número de fallos en base a la siguiente fórmula:

$$Fallos = FP + FN$$

6. Cálculo de la función de fitness en base a la siguiente expresión matemática:

$$Fitness = \begin{cases} Nº\ de\ sujetos + 0{,}1 \times Nº\ de\ variables\ activas / Nº\ de\ variables\ totales\,, \\ si\ la\ regla\ no\ cubre\ ningún\ sujeto\ de\ entrada \\ \\ Fallos\ + (Nº\ de\ sujetos - Aciertos), en\ otro\ caso \end{cases}$$

Criterio de similitud

Dado que en esta aproximación la estructura es menos flexible que en la aproximación 2, existe una mayor probabilidad de que las expresiones obtenidas sean más similares. Por ello, y en base a las pruebas realizadas durante el desarrollo del método, se observó la necesidad de implementar un mecanismo que permitiese comparar la similitud de las reglas candidatas obtenidas para que el conjunto de expresiones finales fuese lo más heterogéneo posible. Se comprobará, pues, antes de agregar una regla candidata al conjunto final de expresiones, que esta no esté contenida ya en dicho conjunto.

2.1.2. Aproximación 2: núcleo basado en PG

Con el objetivo de obtener expresiones más flexibles que representen el conocimiento subyacente en los datos de entrada, se desarrolla una segunda aproximación basada en PG. Dado que a veces las expresiones obtenidas utilizando este tipo de técnica resultan complejas, se ha desarrollado también un algoritmo de análisis que realiza un ranking de las variables en función de su relevancia y que estudia la coincidencia de aparición de variables en parejas y tríos.

Codificación

En este caso, se ha utilizado una representación basada en árboles. Por lo tanto, cada individuo de esta aproximación tendrá el aspecto mostrado en las siguientes figuras. Cada nodo del árbol puede representar un valor, una variable o un operador. Dependiendo del modo de PG que se esté utilizando (expresivo – no expresivo), se considerarán unos operadores u otros (operadores lógicos, suma, resta, multiplicación, división protegida…).

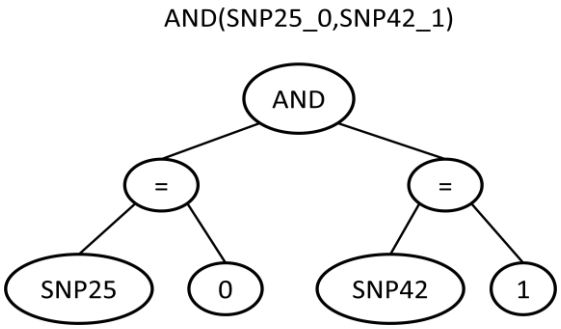


Figura 31. Posible expresión candidata de la aproximación 2a

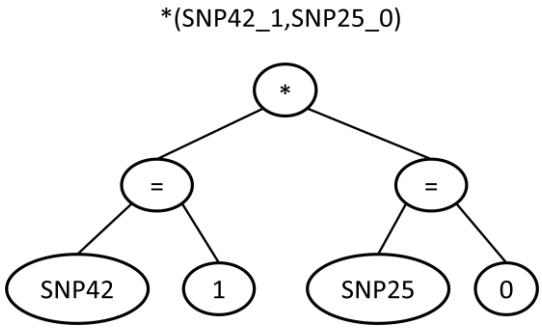


Figura 32. Posible expresión candidata de la aproximación 2b

En todo caso, los individuos deben verse como entes formados por dos partes: un conjunto de variables independientes (mostrados en las figuras anteriores) y una variable dependiente que representa la salida.

Conjunto de terminales y funciones

Aunque existen multitud de funciones que pueden ser utilizadas en PG, normalmente sólo un subconjunto de estas se utiliza de forma simultánea, dado que el tamaño del espacio de búsqueda incrementa de forma exponencial de acuerdo al tamaño del conjunto de funciones. El conjunto de funciones utilizado en este trabajo se detalla a continuación. Dado que las funciones deben cumplir la propiedad de clausura, el operador de división se implementa de forma protegida, es decir, la división protegida funciona exactamente igual que la división excepto cuando el denominador vale cero, en cuyo caso devuelve el valor del numerador.

Aproximación 2a: Modo expresivo

El conjunto de terminales, en este caso, estaría formado por los valores concretos de las variables de cada individuo (significando *variable5_1* que el individuo tiene un 1 en la

variable número 5). El conjunto de funciones estaría compuesto por los operadores mostrados en la Tabla 1. Los individuos expresivos tendrán una salida booleana que puede compararse directamente con la salida esperada al tratarse de 0 ó 1.

Tabla 1. Conjunto de funciones para PG expresivo

Representación	Número de argumentos	Operación
AND	2	Y / Conjunción
OR	2	O / Disyunción
NOT	1	No / Negación
EXOR	2	O exclusivo/ Disyunción exclusiva
=	2	Igualdad
<>	2	Desigualdad

Aproximación 2b: Modo no expresivo

Para este modo, el conjunto de terminales estaría formado por los valores concretos de las variables de cada individuo (significando *variable5_1* que el individuo tiene un 1 en la variable número 5), al igual que el caso anterior, pero, adicionalmente, también se incluyen valores reales en el intervalo [0, 1]. El conjunto de funciones estaría compuesto por los operadores mostrados en la Tabla 2.

Además, en este caso, la salida es un número real, por lo que se aplicará un umbral de 0,5 para decidir si la salida es positiva o negativa.

Tabla 2. Conjunto de funciones para PG no expresivo

Nombre	Número de argumentos	Operación
+	2	Suma aritmética
-	2	Resta aritmética
*	2	Multiplicación aritmética
%	2	División protegida

Función de fitness

Al igual que en la aproximación anterior, la función de fitness fue diseñada de tal forma que se tuviesen en cuenta una serie de factores que, con diferentes pesos establecidos de forma empírica, penalizasen características no deseables, como los falsos positivos o las expresiones demasiado largas. El fitness es calculado siguiendo el algoritmo detallado a continuación:

1. Evaluación de la expresión sobre el conjunto de datos de entrenamiento.
2. Cálculo del número de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN) de aquellos sujetos no cubiertos por las reglas obtenidas por el algoritmo de PG.
3. Cálculo del número de verdaderos negativos y falsos positivos de aquellos sujetos cubiertos por las reglas, con el objetivo de penalizar aquellas reglas que supongan una contradicción.
4. Cálculo de la variable fallos siguiendo la fórmula:

$$Fallos = \frac{12 \times FP + 8 \times FN}{Número\ de\ sujetos}$$

5. Cálculo la variable aciertos siguiendo la fórmula:

$$Aciertos = \frac{0{,}5 \times VP + 0{,}01 \times VN}{Número\ de\ sujetos}$$

6. Cálculo del número de variables que forman parte de la expresión devuelta por el algoritmo de PG.
7. Penalización de expresiones con una única variable o un número muy grande de variables.
8. Cálculo del valor de fitness:

$$Fitness = \begin{cases} Fallos + Penalización - Aciertos, si\ fitness > 0 \\ 0, en\ otro\ caso \end{cases}$$

Análisis de expresiones

Con el objetivo de aportar más información al investigador y simplificar la interpretación de las expresiones obtenidas como resultado de aplicar la PG, se desarrolló un algoritmo de análisis de expresiones basado no sólo en el número de apariciones de cada variable, sino también en diversas medidas estadísticas para dar mayor robustez al método.

A partir de los resultados obtenidos durante la fase de validación del método desarrollado, en primer lugar, se realiza el cálculo del número de apariciones de cada variable en el conjunto de expresiones obtenido por el sistema, además de la media de la precisión (calculada como el número de verdaderos positivos entre el total de clasificados como positivos) y *odds ratio* (definido en el ámbito clínico como la posibilidad de que una condición de salud o enfermedad se presente en un grupo de población frente al riesgo de que ocurra en otro) de las expresiones en las que la variable está presente. Estas medidas complementan al número de apariciones, aportando información de gran relevancia dado que a pesar de que el usuario podría realizar el ranking de forma manual él mismo, además de lo costoso que supone, desconocería los valores de las evaluaciones de las pruebas sobre el conjunto de datos que el programa ha ejecutado. Estas medidas estadísticas, que han demostrado empíricamente ser las más adecuadas de entre todas las probadas, son combinadas con el número de apariciones con el objetivo de determinar la importancia de la variable dentro de las soluciones y sus pruebas. La importancia relativa de cada variable se calcularía de acuerdo a la siguiente fórmula:

$$Importancia\ variable_i = Precisión \times Odds\ ratio \times \frac{Número\ de\ apariciones(variable_i)}{Número\ total\ de\ expresiones\ obtenidas}$$

Adicionalmente, cuando una pareja o un trío de variables coincidan en un número relevante de expresiones extraídas, se mostrarán al usuario indicando el porcentaje de ocasiones en el que coinciden. Se decidió analizar combinaciones de dos y tres variables puesto que eran las situaciones más comunes y a partir de esta información se pueden obtener, incluso, conclusiones sobre relaciones en las que estén involucradas un número mayor de variables.

Con el objetivo de evitar saturar al usuario con información sobre relaciones que podrían ser puntuales o casuales, se ofrece la posibilidad de fijar umbrales para controlar esta situación. En particular, para las parejas se ha exigido, en el caso al que se ha aplicado el

método desarrollado en esta tesis, que cada una de las variables involucradas esté presente en al menos el 25% de apariciones de la otra para considerar esa pareja como representativa, mientras que en el caso de los tríos se ha bajado este umbral a 15%. Se utiliza, por tanto, para valorar si una relación es representativa el concepto de presencia de una pareja con respecto a cada una de las dos variables, calculada mediante la siguiente fórmula:

$$Presencia\ de\ la\ pareja\ v_i, v_j\ sobre\ v_i = \frac{Número\ de\ apariciones\ conjuntas\ de\ v_i y\ v_j}{Número\ de\ apariciones\ de\ v_i}$$

En base a esta regla, por lo tanto, si dos variables aparecen en, por ejemplo, tan sólo un 10% de las expresiones pero siempre que aparecen están juntas, es probable que exista una relación entre ellas.

2.2. Parámetros

El modelo propuesto en esta tesis acepta una serie de parámetros que pueden ser divididos en dos grupos diferentes:

- Parámetros propios de los algoritmos evolutivos: representan los parámetros configurables tanto en los AA.GG. como en la PG, como la tasa y algoritmo de cruce, la probabilidad y algoritmo de mutación, el tamaño de la población, etc.
- Parámetros globales como el umbral de calidad mínima que deberán cumplir los individuos para ser introducidos en el conjunto final de reglas.

Para que el modelo propuesto esté lo más automatizado posible y no dependa tanto del criterio de un experto, se han realizado numerosos experimentos con el objetivo de probar la eficiencia del sistema y establecer un conjunto de valores para los parámetros anteriormente descritos que ofrezcan buenos resultados.

V. Resultados

1. Aplicación a un caso real: detección de la predisposición genética al desarrollo de la esquizofrenia

1.1. Descripción del problema

La presente tesis ha sido aplicado a un caso real: la extracción de reglas de asociación sobre datos relacionados con la esquizofrenia. Para comprender mejor dicha aplicación, a continuación se explicarán algunos conceptos relacionados con la enfermedad en cuestión.

1.1.1. La esquizofrenia

La esquizofrenia es un síndrome heterogéneo que se caracteriza por perturbaciones del lenguaje, la percepción, el pensamiento, las relaciones sociales y la voluntad. No tiene manifestaciones patognomónicas, es decir, no existe un conjunto de síntomas que caracterice de forma única esta enfermedad. Este síndrome suele aparecer en los últimos años de la adolescencia, su comienzo es insidioso y clásicamente su pronóstico es malo, ya que el paciente progresa desde el retraimiento social y las distorsiones de la percepción hasta un estado crónico de delirios y alucinaciones.

Los enfermos con esquizofrenia pueden presentar síntomas positivos (tales como desorganización conceptual, ideas delirantes o alucinaciones) o síntomas negativos (deterioro funcional, anhedonia, expresión emocional reducida, alteración de la concentración y disminución de la relación social) y deben tener al menos dos de estos síntomas durante un período mínimo de un mes, y síntomas continuos de la enfermedad durante al menos 6 meses para que la enfermedad se pueda diagnosticar formalmente. Sin embargo, varias enfermedades psiquiátricas se pueden presentar de forma parecida y por

ello, inicialmente, a veces es difícil distinguir la esquizofrenia de otros trastornos mentales, además de que no existe un criterio único determinado por los expertos a la hora de diagnosticar la enfermedad.

A pesar de que durante años los investigadores han buscado sin resultado una causa única para la esquizofrenia, la mayoría han concluido en señalar que la esquizofrenia sería la consecuencia de varios efectos acumulativos de determinados factores de riesgo, desglosados *grosso modo* en geneticofamiliares y ambientales (Chinchilla Moreno, 2007).

Diversos estudios en familias, gemelos y adoptados confirmaron y han permitido cuantificar la contribución genética a la esquizofrenia (Sham, 1996). A partir de estos datos se empezaron a aplicar las técnicas de la genética molecular para identificar los genes causantes de la enfermedad (Sáiz and Fañanás, 1998). Estos genes no son los genes de la esquizofrenia en sí mismos, sino que transmitirían un conjunto de características que aumentarían el riesgo de presentar la esquizofrenia.

Como se ha comentado anteriormente, la predisposición a padecer esquizofrenia no viene determinada por un gen, sino que se piensa que es debida a la interacción de varios fragmentos de genes o SNPs.

Una de las principales limitaciones que tienen los estudios de ligamiento en el caso de las enfermedades genéticamente complejas, como la esquizofrenia, es que tienen poco poder para detectar genes de efecto menor. Por ello, en este caso, los datos utilizados fueron obtenidos siguiendo el diseño de un estudio de asociación. En esta tesis, de las aproximaciones mencionadas en el capítulo de "Fundamentos", se ha seguido la de genes candidatos, ya que los datos reales proporcionados eran de unos genes concretos y, además, esta aproximación ofrece la posibilidad de estudiar de forma más precisa las bases genéticas del trastorno en cuestión (Sáiz Ruiz, 1999). Sin embargo, el método desarrollado podría ser aplicado igualmente a GWAs.

1.2. Conjuntos de datos

1.2.1. Datos sintéticos

Con el objetivo de refinar el modelo propuesto, y dado que el esqueleto general ya había probado su buen funcionamiento previamente sobre bases de datos validadas (Aguiar-

Pulido et al., 2010), se generaron 360 conjuntos de datos sintéticos. Estos conjuntos fueron elaborados de tal forma que la aplicación del modelo propuesto sobre ellos fuese suficientemente parecida al caso real estudiado en esta tesis, pero que a su vez se pudiese comprobar el correcto funcionamiento del modelo desarrollado de forma exhaustiva.

Cada conjunto de datos, almacenado en un fichero, contiene 614 ejemplares, los cuales están compuestos por 49 variables independientes, además de una posición adicional (variable dependiente) indicando el tipo de ejemplar (caso o control). Para cada conjunto, el porcentaje de casos y de controles es aproximadamente del 50%. Dichos conjuntos pueden ser divididos en cuatro grupos, para los cuales se modificaron los ejemplares que representaban los casos en un porcentaje distinto, obteniendo así los siguientes grupos: 20%, 40%, 60% y 80%.

Ejemplo: Para el grupo del 80%, como se puede ver en la Figura 33, un 50% de los datos representaban controles y el otro 50% casos. Dentro de los casos, se modificó el 80% de los ejemplares con las reglas generadas de forma aleatoria, es decir, un 40% del total de los casos.

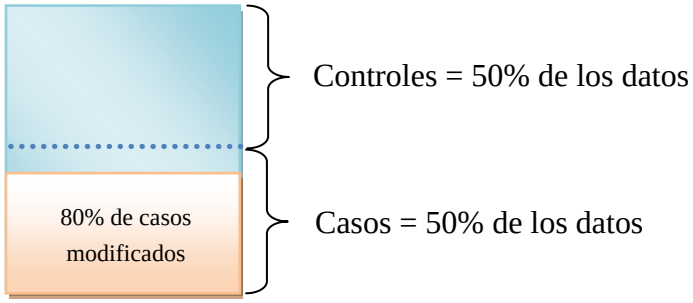


Figura 33. Generación de datos

Cada grupo anterior, como se puede ver en la Figura 34, se divide en tres: los conjuntos de datos modificados por una regla, los conjuntos modificados por dos reglas y los conjuntos modificados por tres reglas. Estos tres conjuntos, a su vez, se dividen en tres: los conjuntos de datos modificados por reglas con dos variables, los conjuntos modificados por reglas con tres variables y los conjuntos modificados por reglas de cuatro variables. Finalmente, dentro de cada grupo de estos se encuentran diez conjuntos de prueba. Así, se tendrán diez conjuntos de datos a los que se ha aplicado una regla con dos variables modificando el 80% de los casos, diez conjuntos de datos a los que se ha aplicado una regla con tres variables modificando el 80% de los casos, etcétera.

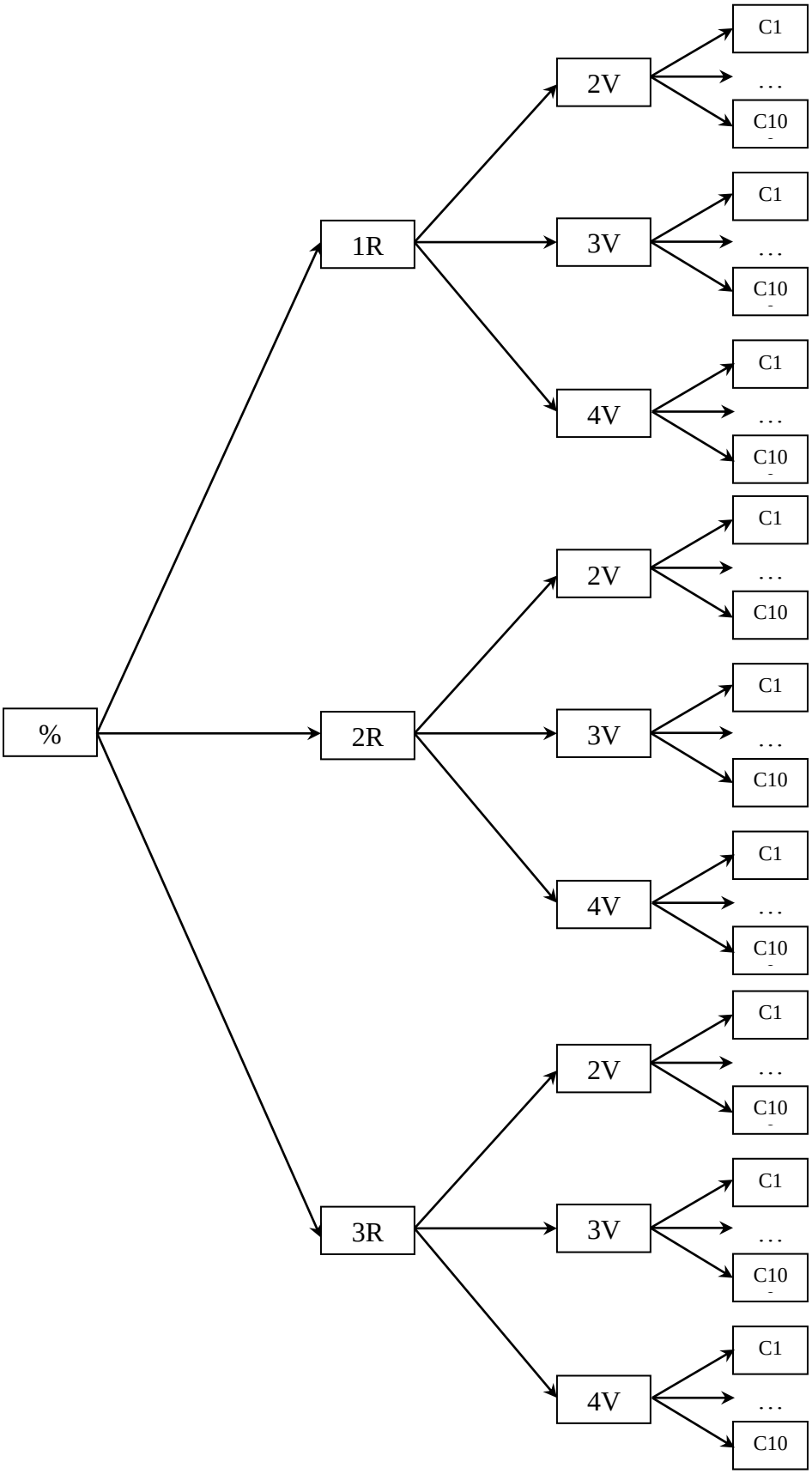


Figura 34. Un grupo de conjunto pruebas

La generación de ficheros de prueba se realizó, pues, siguiendo los siguientes pasos:

1. Se generaron 360 ficheros con datos aleatorios. Cada uno consistía en un conjunto de 614 ejemplos, siendo cada ejemplo una lista de 50 números enteros con valores entre 0 y 2, excepto aquel que representa la clase (control o caso), que sólo podía tomar el valor 0 ó 1 (respectivamente). De este modo, se tendrían representados 49 SNPs más una etiqueta de clase. Además, el 50% de cada fichero contenía casos y el otro 50% controles.

2. Se generaron reglas con 2, 3 ó 4 variables, siguiendo la distribución de probabilidad existente en el fichero de datos clínicos reales para los posibles valores que podía tomar cada SNP.

3. Según el grupo al que pertenecía el fichero, se modificó por 1, 2 ó 3 reglas que contenían 2, 3 ó 4 variables. Para ello, se sustituyeron los valores de los SNPs generados de forma aleatoria por aquellos que contenían las reglas generadas. Esto se hizo para un 20%, 40%, 60% u 80% de los casos, en función del grupo al que perteneciese el fichero. Cada ejemplo sólo podía ser modificado por una única regla.

De este modo, por lo tanto, sería posible verificar si el algoritmo es capaz de encontrar las reglas introducidas de forma artificial.

1.2.1. Datos clínicos reales

El modelo propuesto ha sido aplicado a un conjunto de datos clínicos reales proporcionado por el Grupo de Medicina Xenómica de la USC, liderado por Ángel Carracedo, en colaboración con la *Fundación Pública Galega de Medicina Xenómica*. Estos datos han sido obtenidos de diversos hospitales psiquiátricos gallegos. El conjunto en cuestión contenía 614 ejemplares, cada uno con 49 SNPs pertenecientes a dos genes diferentes (HTR2A y DRD3), además de la correspondiente posición donde se indica el tipo de ejemplar, de los cuales 354 eran controles y 260 eran casos.

Los datos se presentan en un fichero y cada paciente consiste en una lista de números enteros que, salvo la última posición, representan un valor para un SNP, que puede ser: homocigótico para el alelo 1, heterocigótico, homocigótico para el alelo 2 o desconocido. El valor de desconocido se debe a errores al realizar el genotipado, cosa que suele suceder

con bastante frecuencia, y dichos errores se encuentran distribuidos de forma uniforme en los datos. Finalmente, en la última posición de cada sujeto se tendrá un valor que indica si el paciente ha sido diagnosticado como esquizofrénico o no.

2. Parámetros utilizados

A lo largo de la realización de las pruebas descritas en la sección siguiente, se han ido ajustando los parámetros hasta obtener aquellos que permitían obtener mejores resultados. A continuación se muestra la configuración del algoritmo evolutivo que, se determinó, era óptima para las distintas aproximaciones desarrolladas.

2.1. Aproximación 1: núcleo basado en AA.GG.

La configuración final utilizada en los experimentos ha sido la siguiente:

Tabla 3. Parámetros para la aproximación 1

Parámetro	Configuración
Algoritmo de creación	Generación aleatoria de la población
Algoritmo de selección	Torneo
Algoritmo de cruce	2 puntos reales
Tasa de cruce	90%
Algoritmo de mutación	Reemplazo aleatorio
Probabilidad de mutación	1%
Número máximo de generaciones	50
Cambio mínimo	0,00000001
Tamaño de la población	750

2.2. Aproximación 2: núcleo basado en PG

La configuración final utilizada en los experimentos para esta aproximación ha sido la siguiente:

Tabla 4. Parámetros para la aproximación 2

Parámetro	Configuración
Algoritmo de creación	*Ramped half-and-half method*
Algoritmo de selección	Torneo
Tasa de cruce	70%
Algoritmo de mutación	Mutación de subárbol
Probabilidad de mutación	10%
Número máximo de generaciones	5.000
Cambio mínimo	0,00000001
Tamaño de la población	20.000
Elitismo	Sí
Altura máxima	4
Altura inicial	3

3. Resultados

En esta sección se describen los experimentos realizados, en primer lugar, para probar el correcto funcionamiento del modelo propuesto y, posteriormente, su aplicación sobre un caso real, comparando los resultados obtenidos por las dos aproximaciones.

Dada la naturaleza estocástica de los modelos de CE, existe una gran componente aleatoria intrínseca que hace que los resultados obtenidos por dichos modelos puedan llegar a tener una gran variabilidad. Para minimizar este efecto lo máximo posible, cada experimento se ha ejecutado diez veces, obteniendo como resultado final la media de cada medida utilizada calculada a partir de dichas ejecuciones.

3.1. Datos sintéticos

Antes de entrar a valorar los resultados obtenidos por las distintas aproximaciones tras la aplicación de cada una a los 360 conjuntos de datos descritos en la sección anterior, es necesario tener en cuenta que, debido a la naturaleza de los datos, es imposible clasificar de forma correcta el 100% de los ejemplos. A continuación se explicará el por qué de esta

afirmación y cómo afecta, detallando cómo se calcula el porcentaje máximo teórico de clasificación.

Deben considerarse dos tipos de errores:

- El primer tipo se debe a que no se modifican todos los enfermos (o casos). Como se ha explicado al describir la generación de los datos, solamente se modifica un 20%, 40%, 60% u 80% de los casos, que representan un 10%, 20%, 30% ó 40%, respectivamente, de los datos totales.

  Por ejemplo: Si se modifica, con una regla, el 80% de los enfermos (los cuales representan el 40% del total), el 20% restante (que corresponden al 10% del total) no tiene el patrón de la regla, por lo que es imposible que el sistema los clasifique como enfermos o casos. Debido a este error, el máximo teórico de clasificación, en este caso, sería del 90%.

- El segundo tipo se debe a que, por azar, un porcentaje de los ejemplos coincidirá con los valores de las reglas.

  Por ejemplo: Si una regla modifica 2 SNPs y cada SNP tiene 3 valores posibles, al generar todos los ejemplos al azar, uno de cada 9 ejemplos, tanto sanos (controles) como enfermos (casos), tendrá ya los valores que indican la regla para enfermo; es decir, uno de cada 9 del 50% de sanos, si el sistema es capaz de encontrar la regla que se ha aplicado, los clasificará incorrectamente como enfermos. En este caso, esto representaría un 5% del total, por lo que debido a este error, el máximo teórico de clasificación bajaría hasta el 85%.

La Figura 35 muestra de forma gráfica lo explicado anteriormente.

Esta situación se da para cada grupo de conjuntos, variando en función del número de casos modificados y el número de variables que contienen dichas reglas. Dado que se han modificado entre el 20% y el 80% de los casos, esto implica que, en la mejor de las situaciones (es decir, cuando las reglas contienen bastantes variables), como mucho habrá entre 60% y 90% de aciertos.

A continuación se muestran los resultados obtenidos para cada aproximación. Se mostrará en cada caso, mediante gráficas, la media de los porcentajes de clasificación obtenidos para cada grupo de conjuntos de datos (20%, 40%, 60% y 80% de casos modificados),

dependiendo del número de reglas que modificaban dichos datos y del número de variables que contenía cada regla (Figura 36 a Figura 47). También se proporcionan, de forma más detallada, mediante tablas, los resultados de las medias obtenidas para cada tipo de conjunto (Tabla 5 a Tabla 16). Además de los resultados por cada conjunto de datos probado, se presentan las medias para cada grupo de conjuntos modificado por una, dos o tres reglas que contenían dos, tres o cuatro variables.

Figura 35. Ejemplificación del máximo teórico de clasificación

3.1.1. Aproximación 1: núcleo basado en AA.GG.

En primer lugar, se muestran las medias obtenidas para los conjuntos de datos en los que se ha modificado el 20% de los casos. Se puede observar que estos resultados suelen ser mejores cuando se introduce una regla, aunque no de forma significativa, pero sí se nota una clara mejoría cuando las reglas contienen más variables. Esto es algo de esperar ya que se deja menos espacio al azar. La Figura 36 muestra un resumen gráfico de los resultados para este grupo, mientras que la Tabla 5 proporciona resultados más detallados.

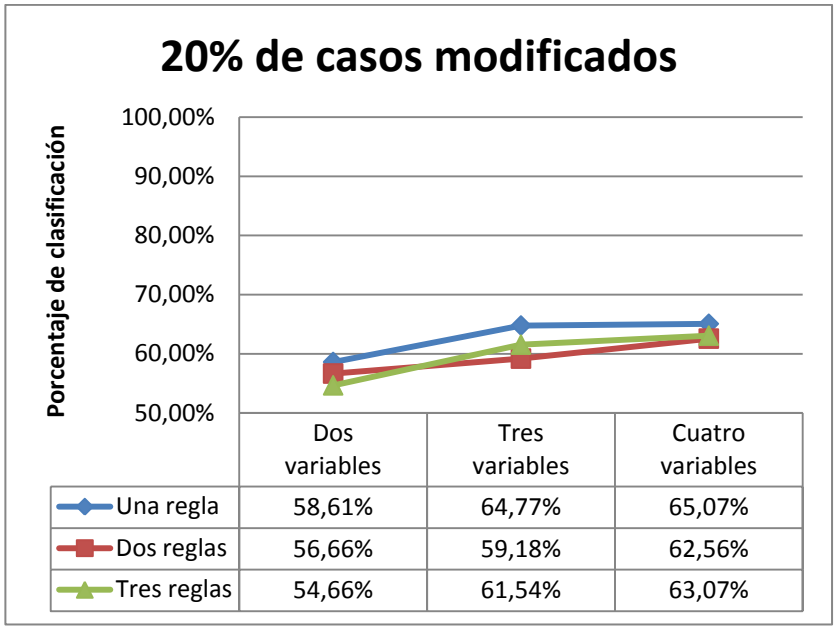


Figura 36. Resultados de la aproximación 1 para conjuntos con el 20% de casos modificados

Tabla 5. Resultados de la aproximación 1 para conjuntos con el 20% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
55,74%	58,85%	75,25%	55,08%	57,38%	77,05%	51,97%	57,38%	51,80%
61,48%	77,38%	63,28%	56,89%	57,54%	55,41%	54,92%	72,79%	65,08%
61,31%	57,54%	77,70%	59,67%	72,30%	56,39%	53,77%	60,33%	55,90%
60,16%	61,15%	75,08%	58,03%	55,08%	56,07%	52,95%	54,92%	66,23%
57,05%	59,18%	64,92%	51,80%	51,64%	67,38%	54,75%	51,64%	84,10%
55,25%	61,48%	62,13%	57,38%	57,70%	73,77%	58,85%	66,39%	53,44%
59,18%	59,02%	59,84%	60,16%	53,11%	58,36%	56,39%	58,20%	60,33%
59,51%	75,08%	58,69%	55,90%	59,84%	57,21%	52,13%	70,33%	68,03%
58,69%	62,13%	59,34%	57,54%	56,89%	69,34%	54,75%	53,77%	71,15%
57,70%	75,90%	54,43%	54,10%	70,33%	54,59%	56,07%	69,67%	54,59%
58,61%	*64,77%*	*65,07%*	*56,66%*	*59,18%*	*62,56%*	*54,66%*	*61,54%*	*63,07%*

Para los conjuntos de datos con un 40% de casos modificados, la media de los porcentajes de clasificación aumenta, aunque no existe una diferencia tan significativa con respecto al número de variables que contiene cada regla. Esto es lógico, dado que hay un mayor porcentaje de ejemplos modificados. Para este grupo de conjuntos, sin embargo, se hace más patente la diferencia entre que se aplique una o tres reglas a la hora de modificar los

datos. Al igual que para el caso anterior, se muestra un resumen gráfico de los resultados (Figura 37) y resultados más detallados presentados en la Tabla 6.

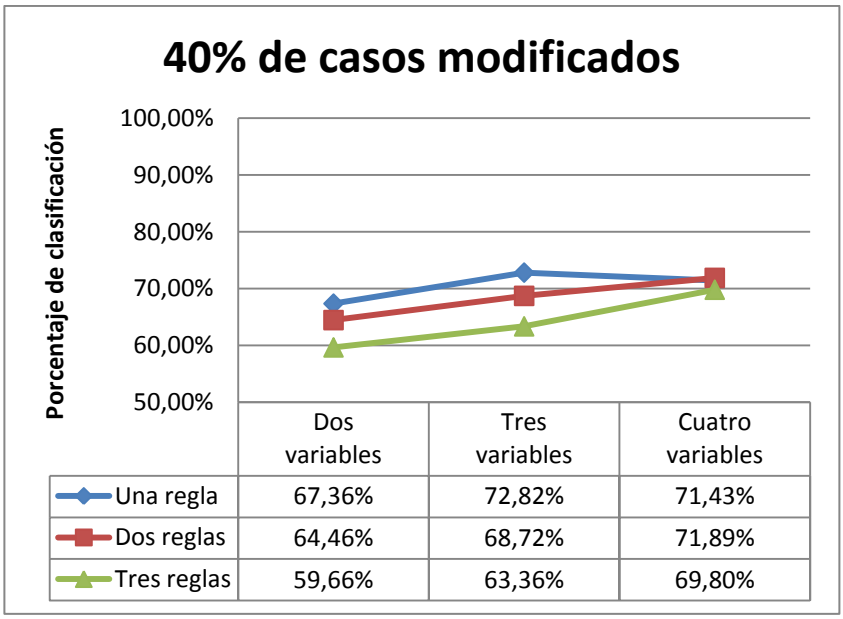


Figura 37. Resultados de la aproximación 1 para conjuntos con el 40% de casos modificados

Tabla 6. Resultados de la aproximación 1 para conjuntos con el 40% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
67,54%	71,80%	79,67%	59,84%	66,56%	64,10%	55,25%	58,69%	88,52%
70,16%	81,15%	66,89%	63,44%	64,59%	68,36%	58,20%	68,69%	69,18%
69,51%	70,49%	80,49%	65,57%	66,56%	70,49%	59,34%	75,57%	64,26%
69,51%	68,69%	79,34%	62,62%	80,33%	76,23%	62,13%	57,38%	77,21%
67,05%	68,69%	66,56%	65,25%	71,15%	74,43%	59,18%	63,44%	60,66%
65,25%	69,51%	68,69%	67,70%	61,80%	76,56%	59,34%	56,89%	62,13%
68,20%	69,18%	69,67%	66,56%	62,79%	70,82%	60,16%	63,77%	85,41%
65,74%	78,03%	69,51%	60,82%	65,41%	76,89%	59,84%	59,18%	62,30%
65,08%	67,70%	68,52%	68,03%	76,07%	74,10%	63,28%	59,67%	61,48%
65,57%	82,95%	64,92%	64,75%	71,97%	66,89%	59,84%	70,33%	66,89%
67,36%	*72,82%*	*71,43%*	*64,46%*	*68,72%*	*71,89%*	*59,66%*	*63,36%*	*69,80%*

Se puede observar cómo a medida que aumenta el número de casos modificados también lo hace el porcentaje de clasificación, haciéndose más clara la diferencia de los porcentajes

de clasificación entre grupos de conjuntos modificados por distintos números de reglas. A continuación se muestran los resultados de clasificación para grupos de conjuntos de datos con el 60% de casos modificados. Dichos resultados, una vez más, se muestran de forma resumida mediante una gráfica (Figura 38) y con mayor detalle en la Tabla 7.

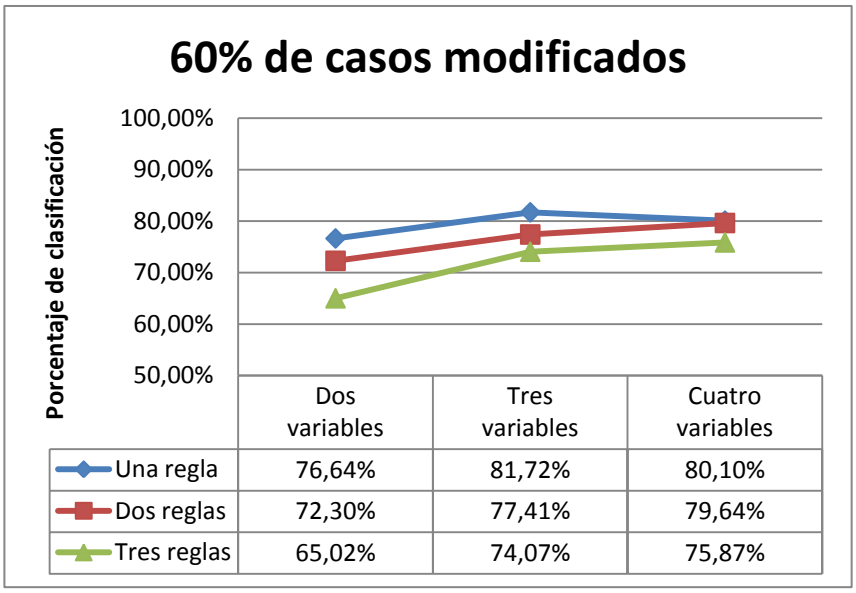


Figura 38. Resultados de la aproximación 1 para conjuntos con el 60% de casos modificados

Tabla 7. Resultados de la aproximación 1 para conjuntos con el 60% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
72,95%	80,00%	82,95%	70,33%	79,18%	82,95%	61,31%	70,98%	72,46%
76,23%	85,90%	77,05%	67,87%	75,57%	77,21%	60,66%	78,03%	78,36%
78,03%	79,67%	87,87%	73,11%	85,41%	80,33%	63,28%	66,39%	69,84%
79,84%	78,20%	82,95%	70,33%	74,75%	65,41%	67,38%	63,44%	77,21%
74,92%	79,51%	79,02%	71,31%	74,59%	86,07%	66,56%	77,05%	86,39%
76,07%	79,84%	80,98%	74,43%	79,51%	82,95%	67,70%	77,70%	71,97%
79,51%	79,84%	78,85%	74,59%	68,52%	81,15%	62,62%	72,62%	72,30%
75,57%	87,38%	77,87%	69,51%	73,11%	78,52%	66,39%	80,49%	79,84%
75,41%	79,51%	77,87%	76,56%	79,67%	83,77%	65,25%	69,84%	80,49%
77,87%	87,38%	75,57%	74,92%	83,77%	78,03%	69,02%	84,10%	69,84%
76,64%	*81,72%*	*80,10%*	*72,30%*	*77,41%*	*79,64%*	*65,02%*	*74,07%*	*75,87%*

Finalmente, se muestran los resultados obtenidos para el último grupo de conjuntos de datos (Figura 39 y Tabla 8).

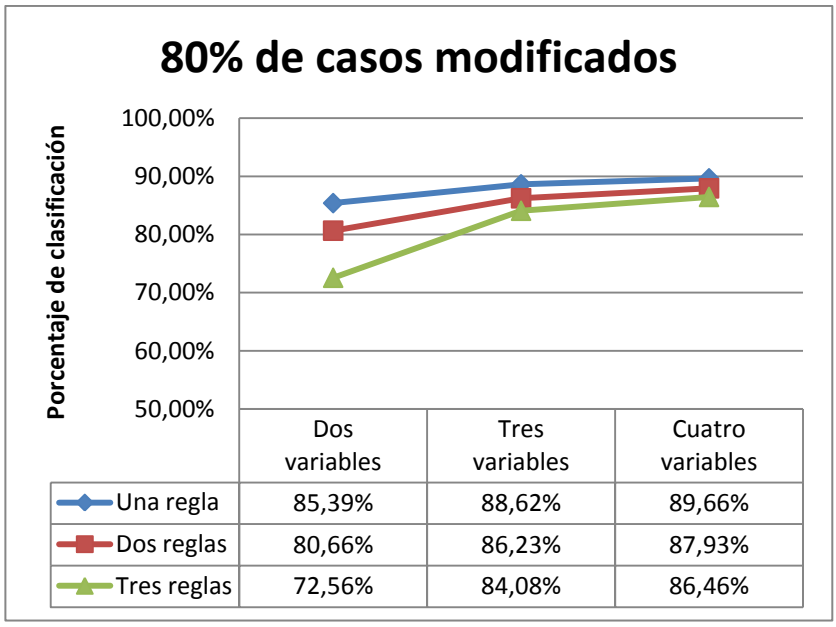


Figura 39. Resultados de la aproximación 1 para conjuntos con el 80% de casos modificados

Tabla 8. Resultados de la aproximación 1 para conjuntos con el 80% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
84,59%	88,03%	87,05%	79,18%	86,56%	88,03%	68,20%	86,07%	84,43%
86,72%	87,87%	89,34%	81,97%	85,74%	89,18%	67,54%	87,38%	91,80%
87,70%	92,13%	90,98%	81,15%	86,72%	89,18%	75,74%	85,08%	91,15%
86,72%	88,03%	90,00%	78,03%	86,07%	76,23%	75,41%	86,07%	84,43%
83,77%	87,21%	89,18%	80,00%	82,62%	92,13%	71,48%	82,62%	80,49%
84,43%	89,02%	91,15%	80,00%	88,52%	85,25%	70,33%	77,38%	84,10%
87,21%	85,57%	88,69%	80,82%	87,21%	92,13%	76,07%	84,75%	85,08%
85,08%	89,84%	91,97%	83,77%	87,87%	91,15%	73,28%	87,21%	87,05%
84,10%	87,70%	89,02%	80,82%	84,43%	88,03%	73,11%	87,54%	86,07%
83,61%	90,82%	89,18%	80,82%	86,56%	88,03%	74,43%	76,72%	90,00%
85,39%	*88,62%*	*89,66%*	*80,66%*	*86,23%*	*87,93%*	*72,56%*	*84,08%*	*86,46%*

3.1.1. Aproximación 2: núcleo basado en PG

Para esta aproximación se han probado los dos modos que ofrece la PG, el modo expresivo y el no expresivo. Observando los resultados de clasificación obtenidos por ambos modos se puede concluir que, aunque no parece haber una diferencia significativa entre ambos modos, para el conjunto de datos con un 60% de casos modificados el modo no expresivo parece obtener mejores resultados que el expresivo.

Modo expresivo

A continuación se muestran los resultados obtenidos para los distintos grupos de conjuntos de datos cuando esta aproximación se aplicó utilizando el modo expresivo del algoritmo de PG (Figura 40 a Figura 43 y Tabla 9 a Tabla 12).

En general, la tendencia es que aquellos conjuntos de datos que han sido modificados por una sola regla obtengan mejores resultados y, especialmente cuando esta aproximación con esta configuración se aplica a conjuntos de datos modificados por dos o tres reglas, se observa una tendencia ascendente en los porcentajes de clasificación cuanto mayor es el número de variables que forman parte de dichas reglas. Como ya se mencionó anteriormente, esto tiene lógica puesto que se deja menos espacio al azar.

La Figura 40 muestra un resumen gráfico de los resultados para el grupo de conjuntos de datos con el 20% de casos modificados, mientras que la Tabla 9 proporciona resultados más detallados.

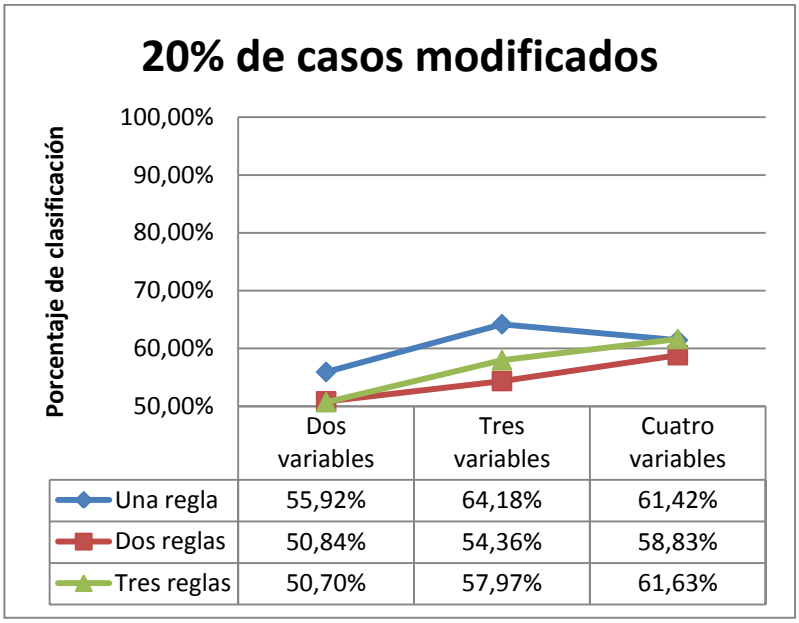


Figura 40. Resultados de la aproximación 2a para conjuntos con el 20% de casos modificados

Tabla 9. Resultados de la aproximación 2a para conjuntos con el 20% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
53,26%	60,16%	73,93%	46,19%	53,62%	66,89%	49,52%	54,48%	45,37%
58,36%	76,23%	53,14%	51,02%	51,21%	48,09%	47,12%	67,21%	66,72%
60,98%	60,16%	75,41%	48,59%	69,02%	54,90%	48,98%	49,99%	54,56%
58,69%	59,18%	70,33%	51,97%	48,27%	53,74%	54,07%	52,68%	62,79%
56,56%	59,02%	60,00%	51,09%	49,67%	68,20%	50,35%	45,92%	86,39%
48,60%	61,15%	59,67%	51,87%	50,84%	70,16%	54,63%	66,07%	56,23%
58,52%	59,18%	57,87%	54,75%	51,95%	49,24%	49,54%	52,79%	58,69%
57,21%	74,59%	54,75%	47,46%	49,35%	53,88%	51,66%	67,05%	70,33%
52,44%	59,84%	55,83%	54,82%	50,47%	67,38%	50,05%	55,62%	66,07%
54,60%	72,30%	53,32%	50,62%	69,18%	55,81%	51,07%	67,87%	49,19%
55,92%	*64,18%*	*61,42%*	*50,84%*	*54,36%*	*58,83%*	*50,70%*	*57,97%*	*61,63%*

Para esta aproximación, la diferencia entre los porcentajes de clasificación obtenido por conjuntos modificados por una regla y el resto se observa incluso antes que en la aproximación anterior. Al igual que para el grupo de conjuntos anteriores, se muestran los resultados obtenidos para aquellos con el 40% de casos modificados de forma resumida (Figura 41) y más detallada (Tabla 10).

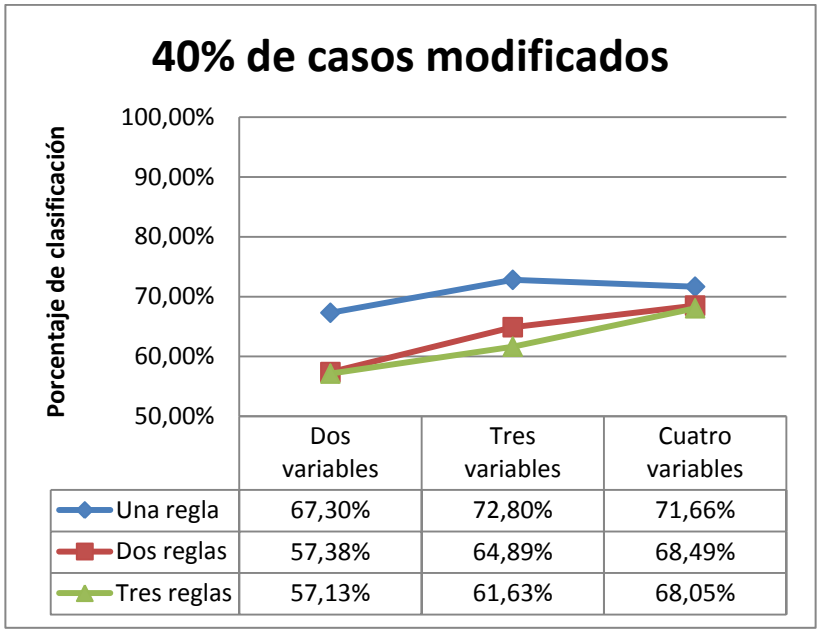


Figura 41. Resultados de la aproximación 2a para conjuntos con el 40% de casos modificados

Tabla 10. Resultados de la aproximación 2a para conjuntos con el 40% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
64,59%	71,15%	80,33%	51,63%	59,67%	68,52%	55,52%	55,08%	87,05%
71,48%	79,51%	68,52%	57,05%	58,52%	58,36%	60,49%	71,31%	70,16%
68,85%	69,84%	80,16%	56,58%	59,84%	61,80%	53,87%	71,64%	63,11%
71,15%	67,87%	79,18%	55,71%	75,25%	74,92%	59,18%	58,36%	71,48%
65,74%	71,48%	68,85%	56,39%	66,89%	77,05%	54,55%	60,16%	60,82%
64,43%	70,00%	72,62%	61,31%	56,72%	72,30%	56,39%	60,49%	62,46%
69,51%	66,89%	68,03%	59,51%	56,23%	60,82%	60,16%	60,82%	86,89%
65,90%	80,33%	67,38%	58,69%	61,48%	76,23%	55,70%	56,87%	58,52%
63,93%	69,67%	66,56%	61,15%	76,72%	70,66%	62,13%	55,35%	56,89%
67,38%	81,31%	64,92%	55,74%	77,54%	64,26%	53,35%	66,23%	63,11%
67,30%	*72,80%*	*71,66%*	*57,38%*	*64,89%*	*68,49%*	*57,13%*	*61,63%*	*68,05%*

Como ya se ha mencionado anteriormente, para aquellos conjuntos con un 60% de casos modificados, en esta configuración del algoritmo se han obtenido resultados de clasificación inferiores a lo que cabría esperar. Dichos resultados, una vez más, se muestran de forma resumida mediante una gráfica (Figura 42) y con mayor detalle en la Tabla 11.

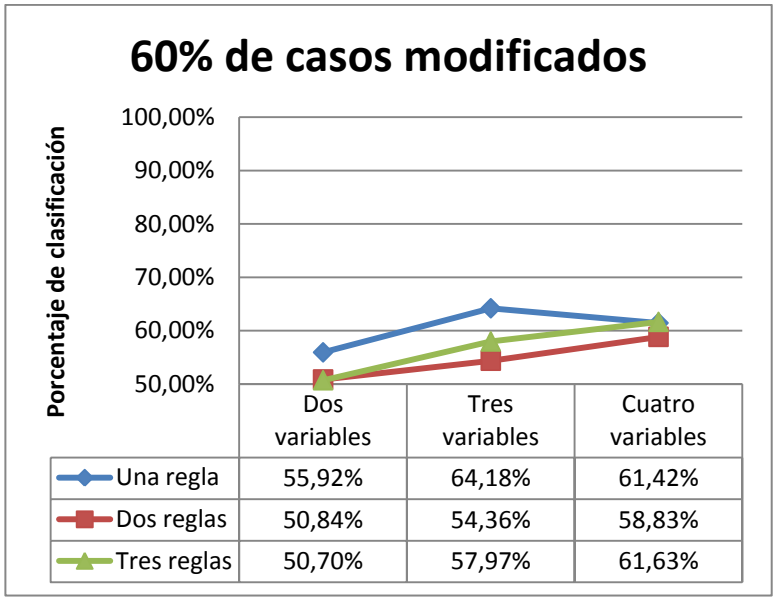


Figura 42. Resultados de la aproximación 2a para conjuntos con el 60% de casos modificados

Tabla 11. Resultados de la aproximación 2a para conjuntos con el 60% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
53,26%	60,16%	73,93%	46,19%	53,62%	66,89%	49,52%	54,48%	45,37%
58,36%	76,23%	53,14%	51,02%	51,21%	48,09%	47,12%	67,21%	66,72%
60,98%	60,16%	75,41%	48,59%	69,02%	54,90%	48,98%	49,99%	54,56%
58,69%	59,18%	70,33%	51,97%	48,27%	53,74%	54,07%	52,68%	62,79%
56,56%	59,02%	60,00%	51,09%	49,67%	68,20%	50,35%	45,92%	86,39%
48,60%	61,15%	59,67%	51,87%	50,84%	70,16%	54,63%	66,07%	56,23%
58,52%	59,18%	57,87%	54,75%	51,95%	49,24%	49,54%	52,79%	58,69%
57,21%	74,59%	54,75%	47,46%	49,35%	53,88%	51,66%	67,05%	70,33%
52,44%	59,84%	55,83%	54,82%	50,47%	67,38%	50,05%	55,62%	66,07%
54,60%	72,30%	53,32%	50,62%	69,18%	55,81%	51,07%	67,87%	49,19%
55,92%	*64,18%*	*61,42%*	*50,84%*	*54,36%*	*58,83%*	*50,70%*	*57,97%*	*61,63%*

Finalmente, se muestran los resultados obtenidos para el último grupo de conjuntos de datos (Figura 43 y Tabla 12).

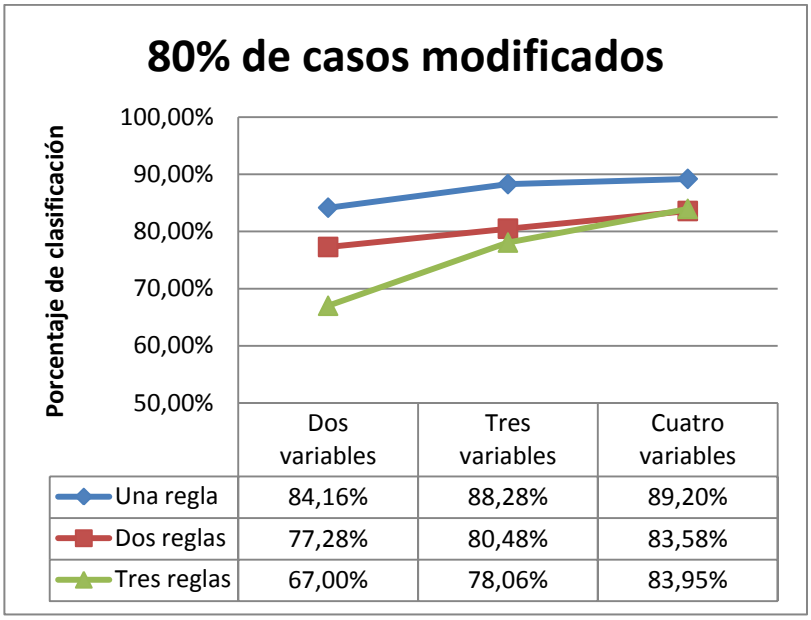


Figura 43. Resultados de la aproximación 2a para conjuntos con el 80% de casos modificados

Tabla 12. Resultados de la aproximación 2a para conjuntos con el 80% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
80,17%	86,39%	87,21%	69,45%	86,23%	83,61%	64,80%	76,81%	81,77%
86,07%	87,21%	89,18%	81,97%	77,06%	82,63%	64,64%	78,93%	90,00%
87,70%	92,62%	90,66%	79,41%	80,19%	84,04%	65,78%	79,30%	89,67%
85,15%	87,38%	89,18%	76,34%	85,41%	88,85%	66,95%	79,73%	79,71%
82,62%	86,39%	89,02%	77,15%	75,88%	83,28%	66,98%	72,20%	81,31%
83,61%	88,03%	90,33%	78,57%	79,97%	81,89%	70,82%	69,24%	80,98%
86,23%	87,21%	87,70%	70,91%	77,26%	81,31%	69,67%	83,77%	82,30%
84,59%	88,85%	91,31%	82,62%	81,22%	86,72%	68,44%	87,21%	89,34%
82,01%	86,23%	88,69%	76,77%	78,16%	84,59%	63,52%	80,33%	81,46%
83,41%	92,46%	88,69%	79,61%	83,44%	78,83%	68,36%	73,05%	82,95%
84,16%	*88,28%*	*89,20%*	*77,28%*	*80,48%*	*83,58%*	*67,00%*	*78,06%*	*83,95%*

Modo no expresivo

A continuación se muestran los resultados obtenidos por la segunda aproximación cuando se ha configurado el algoritmo de PG para que utilice el modo no expresivo (Figura 44 a Figura 47 y Tabla 13 a Tabla 16). Como ya se anunciaba, estos resultados son similares a los obtenidos utilizando la configuración anterior.

La Figura 44 muestra un resumen gráfico de los resultados para el grupo de conjuntos de datos con el 20% de casos modificados, mientras que la Tabla 13 proporciona resultados más detallados.

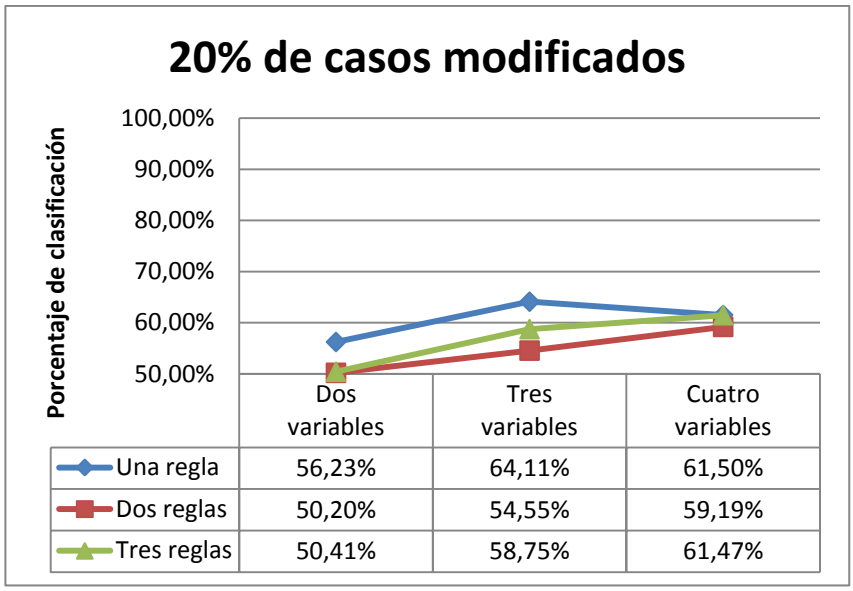


Figura 44. Resultados de la aproximación 2b para conjuntos con el 20% de casos modificados

Tabla 13. Resultados de la aproximación 2b para conjuntos con el 20% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
52,29%	58,33%	73,93%	48,27%	52,07%	69,02%	48,20%	52,46%	49,55%
59,67%	75,90%	56,07%	50,63%	52,04%	50,53%	47,62%	70,49%	67,05%
60,16%	57,87%	75,41%	50,05%	68,36%	54,49%	45,47%	48,45%	55,08%
60,05%	58,69%	66,39%	49,95%	46,96%	54,42%	53,11%	53,58%	64,26%
57,21%	59,67%	59,84%	50,64%	48,18%	66,07%	52,77%	47,71%	86,39%
49,88%	60,82%	59,67%	50,07%	54,56%	66,07%	55,19%	68,69%	52,17%
57,70%	59,34%	59,67%	50,56%	50,94%	54,45%	51,45%	56,67%	56,89%
57,21%	73,77%	54,75%	48,54%	50,18%	55,86%	51,04%	67,05%	68,69%
54,09%	60,33%	55,83%	53,44%	49,55%	67,05%	48,45%	53,17%	65,90%
54,00%	76,39%	53,48%	49,83%	72,62%	53,92%	50,77%	69,18%	48,73%
56,23%	*64,11%*	*61,50%*	*50,20%*	*54,55%*	*59,19%*	*50,41%*	*58,75%*	*61,47%*

Al igual que para el grupo de conjuntos anteriores, se muestran los resultados obtenidos para aquellos con el 40% de casos modificados de forma resumida (Figura 45) y más detallada (Tabla 14).

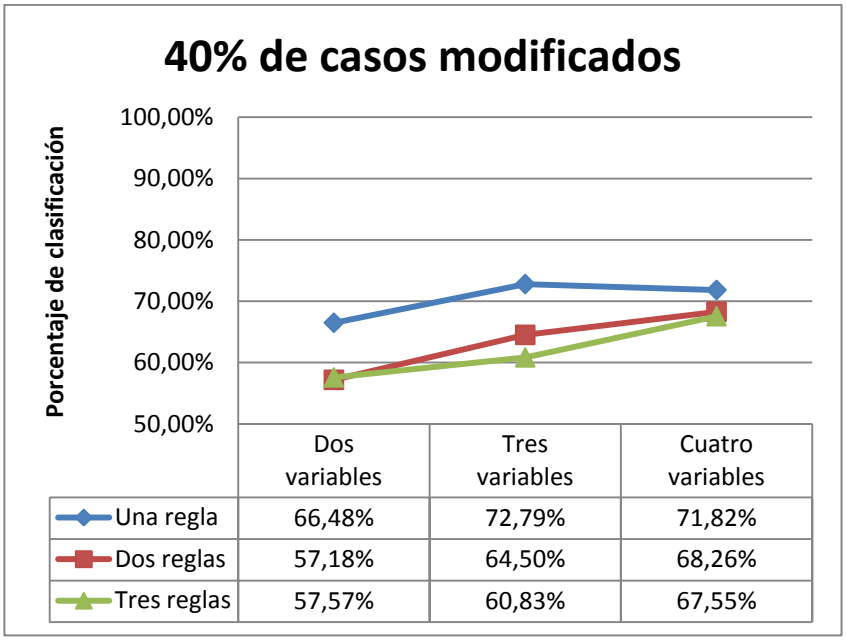


Figura 45. Resultados de la aproximación 2b para conjuntos con el 40% de casos modificados

Tabla 14. Resultados de la aproximación 2b para conjuntos con el 40% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
64,43%	70,82%	81,97%	49,81%	59,67%	66,72%	54,51%	57,38%	86,89%
70,00%	80,16%	67,21%	59,18%	57,03%	63,44%	58,69%	69,02%	67,54%
68,85%	69,18%	81,15%	58,36%	63,11%	61,48%	56,87%	70,98%	63,11%
69,51%	67,87%	79,18%	56,56%	75,25%	74,10%	57,87%	59,34%	72,30%
65,57%	70,49%	69,18%	56,72%	64,75%	76,56%	55,74%	56,56%	58,20%
64,59%	70,00%	72,30%	59,51%	56,39%	72,79%	57,05%	56,56%	61,15%
69,18%	66,39%	69,02%	59,67%	55,66%	61,48%	59,51%	61,31%	86,89%
65,41%	80,66%	65,41%	55,74%	60,16%	74,43%	59,18%	59,18%	56,46%
61,31%	70,00%	66,23%	59,51%	75,25%	68,69%	61,15%	53,42%	59,02%
65,90%	82,30%	66,56%	56,72%	77,70%	62,95%	55,15%	64,59%	63,93%
66,48%	*72,79%*	*71,82%*	*57,18%*	*64,50%*	*68,26%*	*57,57%*	*60,83%*	*67,55%*

En este caso los porcentajes de clasificación van mejorando a medida que el porcentaje de casos modificados incrementa. A diferencia de la configuración anterior, para el grupo de conjuntos de datos con un 60% de casos modificados se continúa en esta línea y se llega incluso a clasificar, de media, el 80,18% de ejemplos. Dichos resultados, una vez más, se

muestran de forma resumida mediante una gráfica (Figura 46) y con mayor detalle en la Tabla 15.

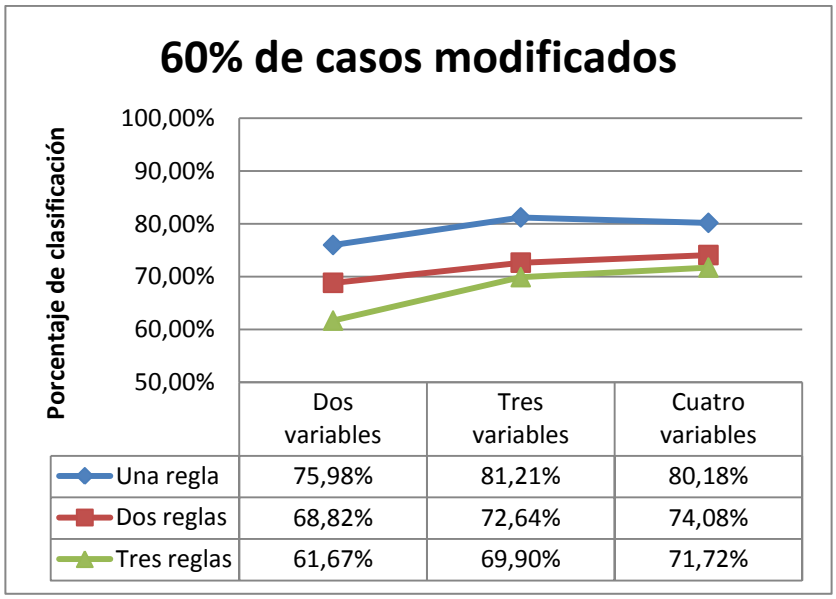


Figura 46. Resultados de la aproximación 2b para conjuntos con el 60% de casos modificados

Tabla 15. Resultados de la aproximación 2b para conjuntos con el 60% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
72,30%	78,69%	87,70%	62,79%	72,79%	81,64%	57,38%	66,72%	63,44%
77,05%	85,57%	76,56%	66,39%	71,80%	66,72%	59,84%	74,26%	74,43%
77,38%	79,34%	87,21%	73,28%	79,51%	75,57%	64,75%	63,61%	66,07%
78,85%	78,03%	85,41%	67,21%	69,02%	71,48%	60,33%	62,62%	72,95%
73,11%	79,18%	75,90%	70,00%	67,21%	78,03%	67,21%	70,33%	89,34%
75,90%	79,67%	80,49%	65,57%	75,90%	76,39%	65,08%	76,23%	62,79%
78,03%	78,85%	76,72%	73,77%	74,59%	71,80%	57,54%	68,69%	68,69%
75,41%	86,56%	77,38%	71,15%	66,39%	69,51%	58,20%	73,44%	75,57%
75,25%	78,85%	78,03%	71,64%	70,49%	78,52%	63,11%	66,23%	78,20%
76,56%	87,38%	76,39%	66,39%	78,69%	71,15%	63,28%	76,89%	65,74%
75,98%	*81,21%*	*80,18%*	*68,82%*	*72,64%*	*74,08%*	*61,67%*	*69,90%*	*71,72%*

Finalmente, se muestran los resultados obtenidos para el último grupo de conjuntos de datos (Figura 47 y Tabla 16).

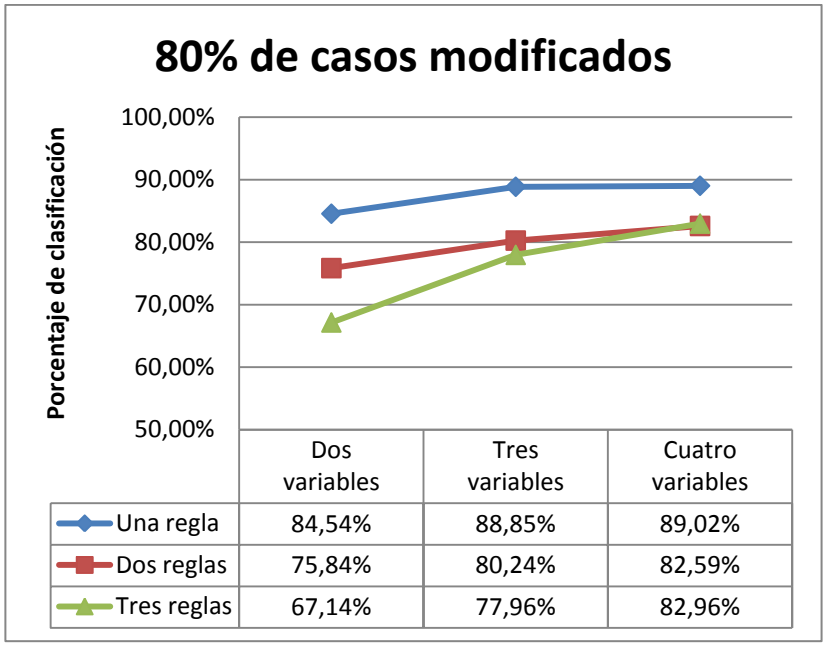


Figura 47. Resultados de la aproximación 2b para conjuntos con el 80% de casos modificados

Tabla 16. Resultados de la aproximación 2b para conjuntos con el 80% de casos modificados

Una regla			Dos reglas			Tres reglas		
2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables	2 Variables	3 Variables	4 Variables
84,10%	87,38%	86,56%	74,45%	87,38%	82,22%	65,43%	77,93%	82,96%
85,57%	87,54%	88,69%	76,26%	82,58%	77,70%	64,40%	77,59%	89,67%
87,05%	92,30%	90,98%	72,99%	81,80%	85,37%	64,92%	78,08%	90,66%
86,72%	87,70%	89,18%	75,36%	81,43%	85,41%	69,09%	77,18%	75,76%
82,13%	86,72%	88,85%	78,43%	76,47%	84,10%	65,73%	72,47%	79,51%
83,61%	88,20%	90,33%	73,39%	78,21%	78,76%	70,31%	68,84%	78,69%
86,23%	88,85%	88,03%	77,05%	76,98%	82,79%	70,28%	84,59%	80,98%
84,92%	89,67%	91,64%	82,62%	78,94%	86,07%	65,30%	87,38%	88,85%
82,46%	87,38%	86,89%	70,58%	75,01%	80,14%	66,86%	81,80%	80,91%
82,62%	92,79%	89,02%	77,24%	83,63%	83,31%	69,09%	73,71%	81,64%
84,54%	*88,85%*	*89,02%*	*75,84%*	*80,24%*	*82,59%*	*67,14%*	*77,96%*	*82,96%*

3.2. Datos clínicos reales

En el caso de datos clínicos reales, se relajó el parámetro que controlaba el porcentaje de ejemplos bien clasificados ya que no se sabe de antemano qué porcentaje de sujetos podrían cubrir las reglas. El objetivo último de aplicar este tipo de modelos a datos clínicos

reales es tratar de generar conocimiento útil para el clínico. Es decir, generar hipótesis acerca de qué variables, en este caso SNPs, podrían estar involucradas en la predisposición genética a desarrollar la enfermedad. Por lo tanto, cuanto más sencillas sean las expresiones obtenidas, más interesante será para el clínico.

A continuación se muestran los resultados obtenidos por las distintas aproximaciones del modelo propuesto en esta tesis, comparados en base a las siguientes medidas:

- Porcentaje de clasificación: porcentaje de ejemplos correctamente clasificados. Se calcula de acuerdo a la fórmula:

$$Porcentaje\ de\ clasificación = \frac{VP + VN}{VP + FP + VN + FN} \times 100$$

- Especificidad: proporción de sanos correctamente identificados. Es decir, la especificidad caracteriza la capacidad de la prueba para detectar la ausencia de la enfermedad en sujetos sanos. Se calcula de acuerdo a la fórmula:

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

- Sensibilidad: proporción de enfermos correctamente identificados. Es decir, la sensibilidad caracteriza la capacidad de la prueba para detectar la enfermedad en sujetos enfermos. Se calcula de acuerdo a la fórmula:

$$Sensibilidad = \frac{VN}{VN + FP}$$

- Valor predictivo positivo: probabilidad de tener la enfermedad si el resultado de la prueba diagnóstica es positivo. Se calcula de acuerdo a la fórmula:

$$(PV+) = \frac{VP}{FP + VP}$$

- Valor predictivo negativo: probabilidad de no tener la enfermedad si el resultado de la prueba diagnóstica es negativo. Se calcula de acuerdo a la fórmula:

$$(PV-) = \frac{VN}{FN + VN}$$

- Odds ratio (OR): posibilidad de que una condición de salud o enfermedad se presente en un grupo de población frente al riesgo de que ocurra en otro. Se calcula de acuerdo a la fórmula:

$$OR = \frac{VP/FN}{FP/VN}$$

En la aplicación a datos clínicos reales, la segunda aproximación supera en porcentaje de clasificación a la primera, pero la primera obtiene valores más altos en cuanto a la especificidad, valor predictivo positivo y OR. Se podría optar por la *aproximación 2b*, ya que es la que mejor clasifica y obtiene resultados aceptables para el resto de métricas analizadas. Estos resultados quedan reflejados en la tabla incluida a continuación.

Tabla 17. Resultados del modelo propuesto al aplicarlo a datos clínicos reales

	Aproximación 1	Aproximación 2a	Aproximación 2b
Porcentaje de clasificación	60,65%	72,28%	72,43%
Especificidad	0,95	0,79	0,85
Sensibilidad	0,14	0,28	0,28
Valor predictivo positivo	0,68	0,50	0,58
Valor predictivo negativo	0,60	0,60	0,62
Odds ratio	3,27	1,5	2,25

4. Discusión

Si se comparan los resultados obtenidos al aplicar las distintas aproximaciones desarrolladas del modelo propuesto, se puede observar cómo la primera aproximación ofrece mejores resultados de clasificación que la segunda en cualquiera de sus variantes tras realizar las pruebas sobre los 360 conjuntos de datos sintéticos (Tabla 18).

Se han aplicado test estadísticos para comparar la primera aproximación con las distintas variantes de la segunda. Se ha estudiado, en primer lugar, si se podía considerar que las varianzas eran iguales, lo cual puede ser asumido, para α=0,05, con un valor de p = 0,11 si se compara la *aproximación 1* con la *aproximación 2a*, y p=9,28 si se compara la *aproximación 1* con la *aproximación 2b*. A continuación se procedió a realizar un *t-test* para determinar si la diferencia en los porcentajes de clasificación era significativa, asumiendo varianzas iguales,

quedando demostrado, para α=0,05, que esta no era significativa (p = 2,44 en el primer caso y p = 2,45 en el segundo). Por lo tanto, se puede concluir que, para este caso, ambas aproximaciones son igualmente válidas.

Tabla 18. Comparativa entre aproximaciones

	Aproximación 1	Aproximación 2a	Aproximación 2b
20%	60,68%	57,32%	57,38%
40%	67,72%	65,48%	65,22%
60%	75,86%	57,32%	72,91%
80%	84,62%	81,33%	81,02%
Media	*72,22%*	*65,36%*	*69,13%*

Con respecto a la aplicación al caso real, escoger entre una aproximación u otra va a depender de qué se valora más. En este caso, dado el tipo de problema, y como se menciona de forma más detallada al comparar el modelo propuesto con otras técnicas similares de minería de datos, suele ser más interesante tener un modelo más específico, ya que la probabilidad de diagnosticar a una persona como enferma de forma errónea decrece cuanto más específico es un modelo, aunque podría optarse por una solución de compromiso y escoger la *aproximación 2b*, ya que obtiene el mejor porcentaje de clasificación y un valor para la especificidad que supera 0,80.

VI. Comparación con otras técnicas

El modelo propuesto en esta tesis, en sus distintas aproximaciones, ha sido comparado con otros métodos de aprendizaje máquina utilizados para realizar clasificación, pero que también extraían expresiones en forma de regla o de árbol, con el objetivo de evaluar su funcionamiento. A partir de la comparación de resultados realizada se extraen una serie de ventajas que el modelo propuesto presenta frente a otras aproximaciones existentes.

1. Método de comparación

A la hora de comparar distintos algoritmos de clasificación, el método más utilizado es el de validación cruzada o *cross validation* para estimar la precisión de los algoritmos. Esta técnica consiste en dividir los datos originales en k subconjuntos, de forma que se utilice un solo subconjunto como datos de validación, mientras que los demás $k - 1$ subconjuntos se utilizan para realizar el entrenamiento (*k-fold cross validation*). El proceso de validación cruzada se repite k veces (el número de *folds*), utilizando una sola vez cada uno de los subconjuntos. Se pueden combinar los k resultados obtenidos de realizar este proceso para obtener un valor global como, por ejemplo, la media de los porcentajes de clasificación.

En esta tesis se ha utilizado la validación cruzada en su modalidad de *10-fold*. La *10-fold cross validation* ha sido un modelo ampliamente aplicado para este tipo de comparativas (Kohavi, 1995b; McLachlan et al., 2004; Picard and Cook, 1984).

2. Técnicas empleadas

Dado que las distintas aproximaciones desarrolladas en esta tesis extraen expresiones en forma de regla o árbol, se ha realizado una comparación con técnicas de minería de datos, también utilizadas en problemas de clasificación e incluidas como parte de una suite ampliamente utilizada, Weka (Hall et al., 2009), que eran capaces de obtener modelos

basados en la extracción de expresiones con una representación similar a la propuesta en este trabajo.

En particular, se han comparado las aproximaciones propuestas con dos grupos de técnicas: métodos de aprendizaje basados en reglas de decisión y métodos de clasificación que utilizan árboles.

2.1. Métodos de aprendizaje basados en reglas de decisión

El objetivo de este tipo de métodos es la generación de un conjunto de reglas de decisión, con el propósito de obtener hipótesis que traten de explicar un determinado sistema.

Para un problema de clasificación, el conjunto de reglas creado tendrá la forma:

$$
\begin{array}{llll}
R_1 & antecedente_1 & \text{then} & c_1 \\
R_2 & antecedente_2 & \text{then} & c_1 \\
 & \ldots & & \\
R_k & antecedente_k & \text{then} & c_1 \\
R_{k+1} & antecedente_{k+1} & \text{then} & c_2 \\
 & \ldots & & \\
R_m & antecedente_m & \text{then} & c_{n-1} \\
 & & \text{else} & c_n
\end{array}
$$

donde R_i es la regla i-ésima, $antecedente_i$ es el conjunto de tests sobre los atributos de entrada, que se realiza para determinar si hay un emparejamiento del ejemplar para su clasificación. Finalmente, cada uno de los c_j se corresponde con la etiqueta de la clase j-ésima.

Con respecto a la sintaxis de las reglas, el antecedente corresponde a una condición, simple o compleja, que se ha de cumplir para que la regla se dispare y, consecuentemente, se seleccione el concepto al que representa. Estas condiciones suelen encontrarse en dos formas distintas: en forma de conjunción de selectores, o en forma de conjunción de literales, correspondientes a predicados de primer orden. De entre estas formas, la primera es la más común. Un selector es una condición de la forma $atributo\ \theta\ valor$, donde θ es un operador relacional, por ejemplo, { >, <, ≥, ≤, = }.

2.1.1. Conjunctive rule

Este método aprende reglas conjuntivas para la predicción sobre datos numéricos y categóricos. Las reglas conjuntivas son aquellas en que los selectores del antecedente se combinan mediante el *AND lógico*.

Para predecir la clase, este clasificador aprende una sola regla. Si esta no cubre al ejemplar que se desea clasificar, entonces, dicho ejemplar, se asociará a la clase asignada a los datos de entrenamiento que no han sido cubiertos por dicha regla (Witten and Frank, 2005).

2.1.1. Decision Table

Este método, dado un conjunto de instancias clasificadas, extraerá unas hipótesis que permitirán clasificar futuras instancias que no estén etiquetadas. Para ello, se basa en las tablas de decisión.

Las tablas de decisión están compuestas por cuatro secciones:

	Reglas de decisión
Identificación de condiciones	Valores de condiciones
Identificación de acciones	Valores de acciones

a) Identificación de condiciones: se detalla una condición por renglón.

b) Identificación de acciones: se describen todos los pasos que se deben realizar. Se llaman acciones a los distintos comportamientos que se asumirán en función de los valores que tomen las condiciones. Se escriben en el orden en que deben ser ejecutadas. En este caso, se tendrá una sola acción posible que será la de clasificar en un conjunto concreto.

c) Valores de condiciones: se indican valores de las condiciones identificadas en la primera sección.

d) Valores de acciones: se indican valores de las acciones descritas en la segunda sección.

Cada columna de la tabla de decisión representa una regla y cada una de ellas, a su vez, constituye una combinación posible de condiciones y de acciones que deben ser tomadas cuando tales condiciones están dadas. Se las enuncia así:

$$SI\ (condición1, condición2, etc.)\ ENTONCES\ (acción1, acción2, etc.)$$

Las tablas de decisión permiten agrupar todas las combinaciones de condiciones y todas las posibilidades lógicas en un conjunto de fácil entendimiento y análisis, creando además la posibilidad de controlar que no se haya omitido ninguna alternativa y que se hayan cubierto todas las posibilidades (Kohavi, 1995a).

2.1.1. JRIP

Este método es capaz de inducir reglas de clasificación a partir de un conjunto de ejemplos, es decir, simplemente construye reglas de la forma si-sino empleando operadores de conjunción y disyunción. Se trata de una extensión de otro algoritmo previo denominado IREP, el cual alterna iterativamente las fases de crecimiento y poda. La poda puede eliminar tanto condiciones en una regla como reglas completas (Cohen, 1995).

2.1.1. NNge

Este método está basado en vecinos cercanos. Utiliza una métrica que mide la distancia entre un nuevo ejemplar y un conjunto de ejemplos en memoria. El nuevo ejemplar se clasificará de acuerdo a la clase de los más cercanos.

Este método, además, utiliza ejemplares generalizados no anidados, los cuales son hiperrectángulos construidos sobre el espacio de características que pueden interpretarse como reglas if-then (Martin, 1995; Roy, 2002).

2.1.1. RIDOR

Inicialmente, genera una primera regla por defecto y, posteriormente, las excepciones. Dichas excepciones son un conjunto de reglas que predicen las clases distintas a la clase por defecto. De esta forma, se generan mediante el algoritmo IREP (base del algoritmo JRIP) y se expanden siguiendo una estructura de árbol (Gaines and Compton, 1995).

2.1.1. DTNB

Corresponde a un clasificador híbrido que construye y utiliza tanto tablas de decisión, como redes bayesianas. En cada punto de la búsqueda, el algoritmo divide los atributos en dos conjuntos disjuntos: uno para la tabla de decisión y otro para la red bayesiana, en este caso *Naive Bayes* (Hall and Frank, 2008).

2.2. Métodos de clasificación que utilizan árboles

Todo árbol de clasificación comienza con un nodo, denominado raíz, al que pertenecen todos los casos de la muestra que se quiere clasificar. El resto de los nodos se dividen en intermedios o no terminales y nodos hoja o terminales, es decir, nodos que ya no se van a dividir más.

En la fase de construcción del árbol, cada nodo hoja se hace corresponder con una categoría concreta de la variable clase. De esta manera, los nodos hoja representan las diferentes particiones en las que se ha dividido el espacio de clasificación. Los nodos que "cuelgan" de un nodo concreto, se dice que son hijos de dicho nodo, y al nodo del que parten las flechas o ramas se le denomina nodo padre.

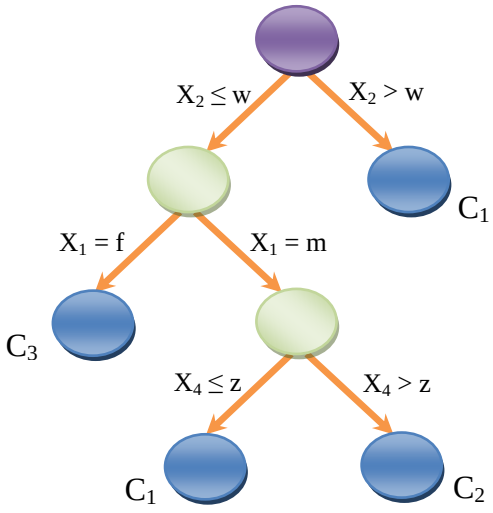


Figura 48. Ejemplo de un árbol de clasificación sencillo

A la hora de clasificar cada patrón, el punto de partida es el nodo raíz y, dependiendo de los valores de la variable por la que se pregunta, los casos se van distribuyendo por los nodos hijo. Este proceso se repite en cada nodo hasta llegar a los nodos hoja.

Se muestra un ejemplo de un árbol de clasificación sencillo en la Figura 48.

2.2.1. J48

Este algoritmo es una implementación del algoritmo C4.5, uno de los algoritmos de minería de datos más utilizado. Dicho algoritmo genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero (depth-first). El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en una mayor ganancia de información (Quinlan, 1993).

2.2.1. REPTree

Árbol de decisión o de regresión que utiliza poda por reducción de error y reducción por ganancia de información (Witten and Frank, 2005).

2.2.1. ADTree

Este método combina árboles de decisión con la precisión predictiva del *boosting*, combinando las hipótesis generadas durante este proceso. Obtiene grafos AND/OR, distribuyendo el conocimiento mediante múltiples ramas a partir de las cuales se realizará la predicción. Se trata de una generalización de los árboles de decisión que combina nodos de decisión y nodos de predicción (Freund and Mason, 1999).

2.2.1. LADTree

Esta técnica (Holmes et al., 2001) aplica un algoritmo de *boosting* logístico para inducir un ADTree, es decir, se trata de un ADTree multiclase que utiliza la estrategia LogitBoost desarrollada por Friedman et al. (Friedman et al., 2000).

2.2.1. NBTree

Este método combina árboles de decisión con clasificadores Naive Bayes en sus hojas (Kohavi, 1996).

3. Resultados obtenidos

A continuación se comparan los métodos presentados con el modelo propuesto. Dado que, para la segunda aproximación, la configuración de PG en modo no expresivo (*aproximación 2b*) ha obtenido mejores resultados, las comparativas se realizarán sobre los resultados de esta aproximación y de la *aproximación 1*.

3.1. Datos sintéticos

En primer lugar, se han comparado los métodos descritos anteriormente con las dos aproximaciones del modelo propuesto, aplicando cada método a los 360 conjuntos de datos sintéticos con el objetivo de realizar pruebas exhaustivas.

Dado que el modelo aquí propuesto se ha comparado con dos tipos de métodos diferentes, se ha estimado conveniente mostrar los resultados por separado.

3.1.1. Métodos de aprendizaje basados en reglas de decisión

A continuación se muestran los resultados obtenidos de comparar las medias de porcentajes de clasificación para cada grupo de conjuntos de datos con distinto porcentaje de modificación de casos, comparando el modelo propuesto con los distintos métodos de aprendizaje basados en reglas de decisión. Se puede observar que la *aproximación 1* del modelo propuesto supera al resto de métodos para todos los grupos de datos (Figura 49). El *eje X* representa los distintos grupos de conjuntos, mientras que el *eje Y* representa los porcentajes medios de clasificación obtenidos.

En la Tabla 19 se muestran, más en detalle, los porcentajes de clasificación medios obtenidos por cada método en función del porcentaje de casos modificados.

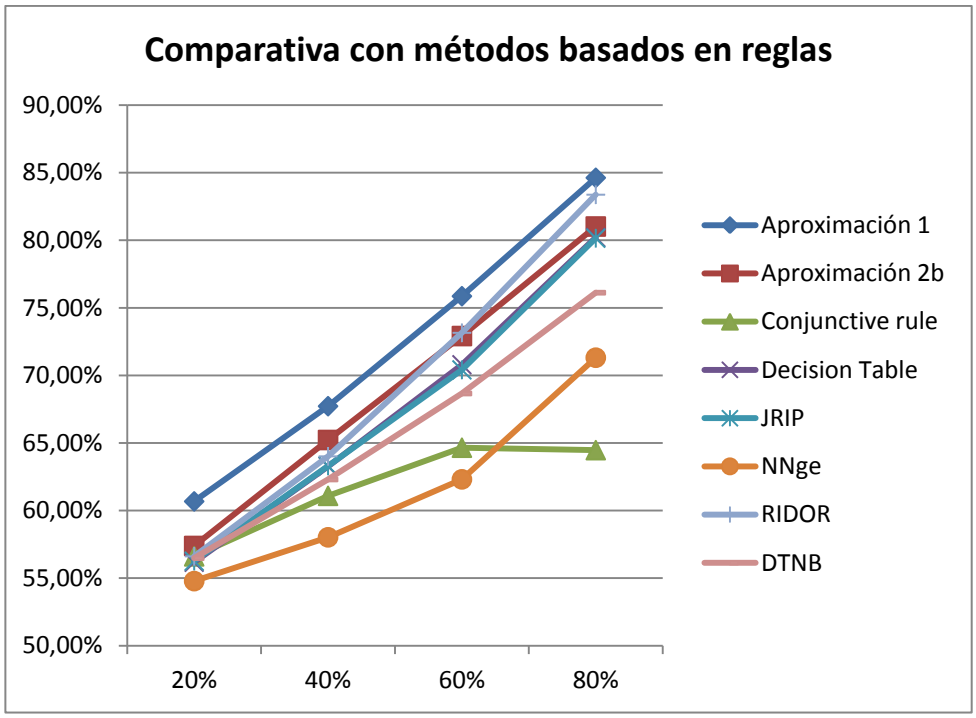


Figura 49. Comparativa con métodos basados en reglas

Tabla 19. Comparativa con métodos basados en reglas

	20%	40%	60%	80%	*Media*
Aproximación 1	**60,68%**	**67,72%**	**75,86%**	**84,62%**	***72,22%***
Aproximación 2b	57,38%	65,22%	72,91%	81,02%	*69,13%*
Conjunctive rule	56,61%	61,09%	64,65%	64,47%	*61,71%*
Decision Table	56,17%	63,27%	70,79%	80,21%	*67,61%*
JRIP	56,26%	63,30%	70,41%	80,15%	*67,53%*
NNge	54,78%	58,02%	62,31%	71,31%	*61,61%*
RIDOR	56,61%	64,01%	73,15%	83,37%	*69,29%*
DTNB	56,28%	62,30%	68,66%	76,12%	*65,89%*

3.1.2. Métodos de clasificación que utilizan árboles

A continuación se muestran los resultados obtenidos de comparar las medias de porcentajes de clasificación para cada grupo de conjuntos de datos con distinto porcentaje de modificación de casos, comparando el modelo propuesto con los distintos métodos de clasificación que utilizan árboles. Se puede observar que la *aproximación 1* del modelo propuesto supera al resto de métodos para todos los grupos de datos y que la *aproximación 2b* los supera también en dos grupos (Figura 50). El *eje X* representa los distintos grupos de conjuntos, mientras que el *eje Y* representa los porcentajes medios de clasificación obtenidos.

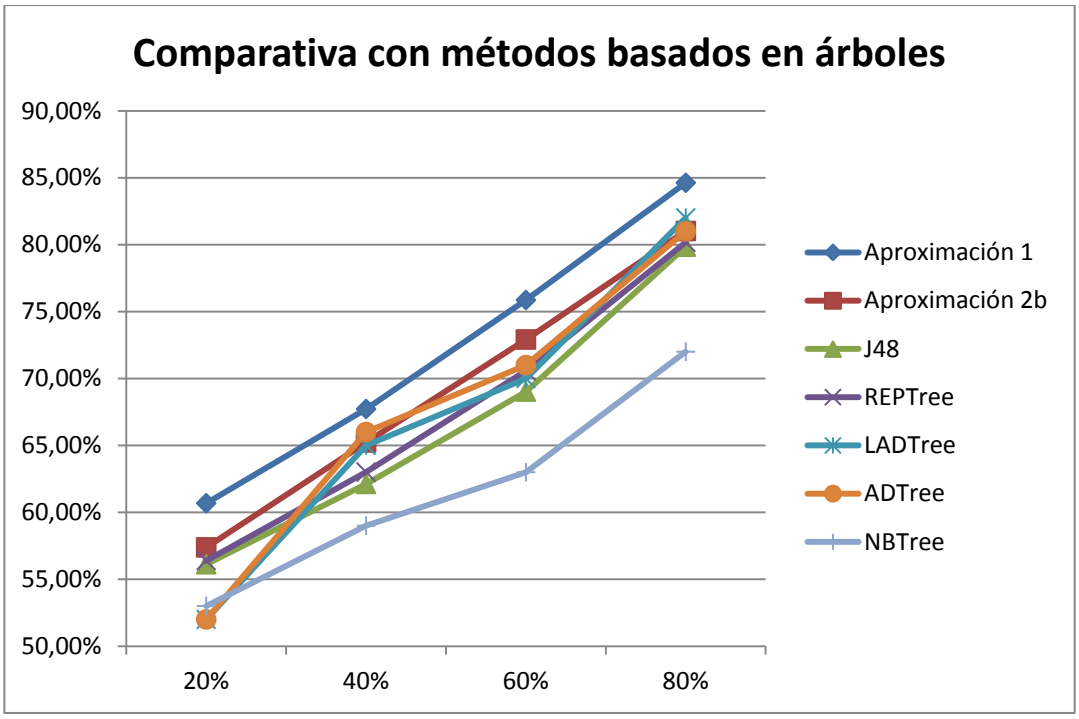


Figura 50. Comparativa con métodos basados en árboles

En la Tabla 20 se muestran, más en detalle, los porcentajes de clasificación medios obtenidos por cada método en función del porcentaje de casos modificados.

Tabla 20. Comparativa con métodos basados en árboles

	20%	40%	60%	80%	*Media*
Aproximación 1	**60,68%**	**67,72%**	**75,86%**	**84,62%**	***72,22%***
Aproximación 2b	57,38%	65,22%	72,91%	81,02%	*69,13%*
J48	56,12%	62,12%	69,02%	79,85%	*66,78%*
REPTree	56,39%	63,01%	70,58%	80,17%	*67,54%*
LADTree	52,00%	65,00%	70,00%	82,00%	*67,25%*
ADTree	52,00%	66,00%	71,00%	81,00%	*67,50%*
NBTree	53,00%	59,00%	63,00%	72,00%	*61,75%*

3.2. Datos reales

El objetivo final de esta tesis es demostrar la utilidad del modelo propuesto en su aplicación a un caso real. Por lo tanto, a continuación se mostrarán los resultados obtenidos tras comparar las dos mejores aproximaciones del modelo propuesto con los distintos métodos anteriores.

La Tabla 21 muestra los resultados de las distintas métricas estudiadas para cada técnica aplicada. Se puede observar cómo la *aproximación 2b* supera claramente en porcentaje de clasificación al resto de técnicas probadas.

4. Discusión

En base a los resultados obtenidos, tanto sobre datos sintéticos como sobre datos clínicos reales, se observa que el modelo propuesto en esta tesis es capaz de obtener mejores resultados de clasificación.

Es interesante destacar también la simplicidad de las expresiones obtenidas por el modelo propuesto y la información adicional que proporciona la *aproximación 2*. A diferencia de otros métodos probados, el número de variables utilizadas en cada expresión es menor.

Es curioso ver también cómo, aplicado a datos clínicos reales, el modelo basado en PG en su modo no expresivo es capaz de clasificar mejor los datos de entrada con una diferencia de casi un 6% con el siguiente mejor método.

Tabla 21. Comparativa de métodos sobre datos clínicos reales

	Clasificación	Especificidad	Sensibilidad	VP+	VP-	OR
Aproximación 1	60,65%	0,95	0,14	0,68	0,60	3,27
Aproximación 2b	72,43%	0,85	0,28	0,58	0,62	2,25
Conjunctive rule	59,61%	0,62	0,54	0,33	0,79	1,87
Decision Table	63,19%	0,64	0,60	0,38	0,82	2,73
JRIP	65,15%	0,68	0,60	0,53	0,74	3,22
NNge	60,26%	0,64	0,54	0,44	0,72	2,04
RIDOR	61,56%	0,61	0,65	0,20	0,92	2,96
J48	66,61%	0,65	0,71	0,35	0,90	4,69
REPTree	62,38%	0,65	0,57	0,46	0,74	2,48
LADTree	60,42%	0,63	0,54	0,41	0,75	2,05
ADTree	64,01%	0,65	0,62	0,38	0,83	3,01
NBTree	64,15%	0,64	0,65	0,34	0,86	3,26

Además, en diagnóstico clínico, cuando el valor de especificidad supera el 80%, se puede considerar bueno. Por regla general, se elige una prueba muy específica cuando se prefiere obtener falsos negativos en lugar de falsos positivos, por ejemplo, para asegurar que un paciente tiene realmente una enfermedad.

Póngase por hipótesis que se tiene un paciente al que es necesario realizarle una prueba cuyo objetivo es determinar la necesidad de operar o no a dicho paciente. En esta situación sería imprescindible asegurarse de que el paciente está enfermo y necesita la operación, dadas las consecuencias que ello conlleva. Otros casos en los que se usa una prueba muy específica involucran casos en que el diagnóstico puede ser una enfermedad grave y prácticamente incurable, cuando es importante desde el punto de vista sanitario y psicológico saber que no se padece la enfermedad o cuando un resultado positivo falso

supone un trauma económico y psicológico para el sujeto. En estas situaciones se utiliza una prueba con un valor predictivo positivo alto que aumenta de valor conforme la prevalencia de la enfermedad es mayor.

Si se analiza el caso aquí estudiado, el modelo que mejor clasifica obtiene un valor de 85% para la especificidad, por lo que probablemente sería el más interesante.

VII. Conclusiones

En este trabajo se presenta un modelo basado en Computación Evolutiva que permite extraer expresiones subyacentes en los datos de entrada que serán utilizadas posteriormente para obtener un modelo de clasificación. Se han desarrollado dos aproximaciones diferentes de dicho modelo: una que utiliza como base AA.GG. y otra que utiliza PG. Estas dos aproximaciones han sido evaluadas de forma exhaustiva para poder analizar su funcionamiento. Ambas han sido aplicadas sobre 360 conjuntos de datos sintéticos, así como a datos clínicos reales, con la finalidad de observar si la forma en que se representan las expresiones extraídas influía sobre los resultados obtenidos por el modelo propuesto. En base a los resultados observados, se puede concluir que no existe una diferencia significativa en cuanto a los porcentajes medios de clasificación obtenidos por ambas aproximaciones, aunque al aplicar dichas aproximaciones sobre datos clínicos reales los modelos obtienen diferencias que deberán ser juzgadas por un experto clínico para escoger cuál se adecúa más en función de los criterios que se quieran valorar más.

La mayor diferencia entre ambas aproximaciones es la forma en que representan las expresiones. En la *aproximación 1* los resultados se obtienen en forma de reglas, mientras que la en la *aproximación 2* estos se ofrecen en forma de árboles. Es de destacar que la *aproximación 2* ofrece un análisis posterior de las expresiones obtenidas, mostrando un ranking que pretende ordenar las variables que forman parte de dichas expresiones en función de su relevancia, además de analizar la coincidencia de pares y tríos de variables. Este análisis aporta información adicional, basada en métricas estadísticas utilizadas en el ámbito clínico, y pretende simplificar la presentación de la información al usuario final.

Se ha realizado una comparación del modelo propuesto en esta tesis con dos grupos de técnicas que obtienen modelos de clasificación basándose bien en expresiones representadas en forma de reglas o bien como árboles, las cuales forman parte de una suite

de minería de datos ampliamente utilizada, Weka, que incluso se incluye como parte de plataformas *opensource* muy reconocidas en el área de *Business Intelligence*, como Pentaho.

El modelo aquí propuesto ofrece una clara ventaja frente al resto, y es que el número de variables utilizadas en las expresiones a partir de las cuales se realizará la clasificación es mucho menor y, por lo tanto, las expresiones obtenidas como resultado final son mucho menos complejas, más abordables y de mayor utilidad para el clínico, además de obtener mejores resultados de clasificación, tanto sobre datos sintéticos como reales, así como un mayor valor para la especificidad.

Finalmente, como resumen y en base a las pruebas realizadas, se puede concluir lo siguiente:

- Como ya se ha mencionado anteriormente, la minería de reglas de asociación puede ser modelada como un problema de optimización combinatoria. En esta tesis, se ha demostrado que el método desarrollado, basado en Computación Evolutiva, es capaz de resolver este tipo de problema de forma satisfactoria, cumpliéndose así la primera hipótesis formulada.
- Mediante la aplicación a un caso clínico real, se ha visto que el modelo desarrollado era capaz de extraer expresiones y de obtener un modelo específico, por lo que esto apoyaría la segunda hipótesis planteada, en la que se enunciaba que la predisposición genética está subyacente en los datos genéticos en algún tipo de estructuras y que las técnicas de Computación Evolutiva son adecuadas para extraer estas estructuras.

Finalmente, se ha visto que cualquiera de las dos representaciones (reglas o árboles) utilizadas en el modelo propuesto han dado resultados satisfactorios, por lo que la elección de una u otra aproximación quedaría supeditada al criterio de un experto en base a si se buscan modelos en donde es más importante la especificidad u otros criterios utilizados en estudios clínicos. Sin embargo, para esta aplicación concreta, se podría concluir que el modelo más adecuado para extraer patrones de predisposición genética a desarrollar la esquizofrenia es aquel basado en PG configurada en modo no expresivo.

VIII. Conclusions

This work presents a model based on Evolutionary Computation which allows extracting underlying expressions in the input data that will be used to build a classification model. Two different approaches have been developed: one based on Genetic Algorithms and another one based on Genetic Programming. These two approaches have been exhaustively evaluated in order to analyze their performance. Both have been applied to 360 generated data sets, as well as to real clinical data, with the aim of determining whether the structure used to represent the expressions extracted had an influence on the results obtained by the model proposed or not. Based on the results, it can be concluded that there is no significant difference between both approaches in terms of mean accuracy. However, the differences between these approaches when applied to real clinical data should be judged by an expert based on the criteria he/she considers more adequate.

The greatest difference between both approaches is the structure used to represent the expressions extracted by the model. *Approach 1* uses rules, while *approach 2* uses trees. It is worth highlighting that *approach 2* provides an additional analysis of the expressions extracted, showing a ranking ordered by relevance, as well as the coincidence of pairs and trios of variables in these expressions. This analysis also provides information based on statistical measures used in the clinical field and tries to simplify the way results are presented to the final user.

The model presented in this thesis has been compared with two groups of techniques capable of building classification models based on expressions represented as rules or trees. These techniques are included in a widely used data mining suite, Weka, which is even provided as part of well-recognized *opensource Business Intelligence* platforms, such as Pentaho.

The model proposed here offers a clear advantage over the others, that is, the number of variables included in the expressions and used to classify is lower. Hence, the expressions

obtained are less complex, more understandable and, thus, more useful for the clinician. In addition, better results in terms of accuracy and specificity are achieved by this model, both using generated and real data.

Finally, to sum up, the following conclusions can be drawn from the work presented in this thesis:

- As mentioned previously, association rule mining can be modeled as a combinatioral optimization problem. It has been proved that the method developed in this thesis, which is based on Evolutionary Computation, is capable of solving this type of problem successfully, confirming, hence, the first formulated hypothesis.
- The method developed has proved, in its application to a real clinical case, that it is able to extract expressions and obtain a specific model. This supports the second formulated hypothesis, which suggested that genetic predisposition is underlying in the data as some type of structure and that Evolutionary Computation techniques are appropriate to extract these structures.

Finally, it has been shown that the structures used (rules or trees) in the proposed model have achieved similar results. Therefore, choosing a specific approach would depend on the expert's criterion, which may be based on whether he/she wants a model with a higher value for specificity or other criteria used in clinical studies. Nevertheless, based on the results of applying the proposed model to the real case presented here, it can be concluded that the most appropriate model to extract patterns of genetic predisposition to develop schizophrenia is that one based on non-expressive Genetic Programming.

IX. Futuras líneas de investigación

El modelo desarrollado en este trabajo ha demostrado ofrecer buenos resultados. A partir de los resultados ofrecidos, el trabajo de investigación podría continuarse en diferentes direcciones.

En primer lugar, teniendo en cuenta la estructura del modelo propuesto, podría estudiarse la posibilidad de optimizar su rendimiento mediante la paralelización de ciertas partes dada la independencia que existe entre ellas. Particularmente en el caso de la aproximación que utiliza como núcleo la PG, para obtener mejores resultados en el menor tiempo posible, se decidió realizar las pruebas en un supercomputador del CESGA. En este caso, el coste computacional se vería reducido de forma considerable. Podría plantearse incluso el uso de computación en *grid* para este propósito.

Otra posible línea de investigación futura en este campo es la adaptación del modelo propuesto para que sea capaz de analizar grandes cantidades de datos catalogados como *Big Data*. Para que el método fuese accesible desde cualquier parte sin carga computacional para el usuario y, enlazándolo con el almacenamiento de gran cantidad de datos, podría verse la posibilidad de ofrecer el modelo propuesto como servicio haciendo uso del *cloud computing*.

Las ontologías permiten anotar semánticamente la información de una fuente de datos, haciendo así posible que el significado de dichos datos pueda ser universalmente entendido por agentes humanos y/o artificiales de forma automática. Es un campo en gran auge en los últimos anos y puede enriquecer notablemente los datos utilizados como entrada, permitiendo extraer más relaciones subyacentes en ellos, que no pueden ser obtenidas a simple vista. En base a esto, como siguiente paso en la evolución de este trabajo de investigación, se estima que el uso de ontologías podría aportar información adicional útil en la resolución de problemas de minería de datos en el ámbito de la biomedicina y, por lo

tanto, ser beneficioso. Por ello, otra posible línea de investigación incluiría la combinación del uso de ontologías con la técnica de minería de datos propuesta, trabajando en lo que se conoce como *ontology-based data mining*.

Finalmente, sería interesante aplicar el modelo presentado en esta tesis a otros casos reales, englobados dentro de las problemáticas relacionadas con fármacos, como la búsqueda de relaciones entre el genotipo de un paciente y posibles efectos adversos de medicamentos.

Referencias

Aguiar-Pulido, V., Seoane, J. A., Freire, A., Guo, L., 2010. GA-based Data Mining applied to genetic data for the diagnosis of complex diseases. Soft Computing Methods for Practical Environmental Solutions: Techniques and Studies. IGI Global, pp. 220-240.

Aguiar, V., Seoane, J. A., Freire, A., Munteanu, C. R., 2009. Data Mining in Complex Diseases Using Evolutionary Computation. Lecture Notes in Computer Science 5517, 917-924.

Aguirre, H. E., Tanaka, K., Sugimura, T., 1999. Cooperative crossover and mutation operators in genetic algorithms Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99. Morgan Kaufmann, Orlando, FL.

Angeline, P. J., 1996. An investigation into the sensitivity of genetic programming to the frequency of leaf selection during subtree crossover. Genetic Programming 1996: Proceedings of the First Annual Conference GP-96. MIT Press, Stanford University, CA, pp. 21-29.

Armitage, P., 1995. Test for linear trends in proportions and frequencies. Biometrics 11, 375-386.

Babovic, V., Keijzer, M., Aguilera, D. R., Harrington, J., 2001a. Automatic Discovery of Settling Velocity Ecuations. Technical Report, D2K-0201-1. Danish Hydraulic Institute.

Babovic, V., Keijzer, M., Aguilera, D. R., Harrington, J., 2001b. An Evolutionary Approach to Knowledge Induction: Genetic Programming in Hydraulic Engineering. Proceedings of the World Water & Environmental Resources Congress, Orlando, FL.

Baker, J. E., 1987. Reducing Bias and Inefficiency in the Selection Algorithm. Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 14-22.

Balding, D., 2006. A tutorial on statistical methods for population association Studies. Nature Review Genetics 7, 781-791.

Ban, H. J., Heo, J. Y., Oh, K. S., Park, K. J., 2010. Identification of type 2 diabetes-associated combination of SNPs using support vector machine. BMC Genet 11, 26.

Banzhaf, W., Beslon, G., Christensen, S., Foster, J. A., Kepes, F., Lefort, V., Miller, J. F., Radman, M., Ramsden, J. J., 2006. Guidelines: From artificial evolution to computational evolution: a research agenda. Nat Rev Genet 7, 729-35.

Beasly, D., Bull, D. R., Martin, R. R., 1993. A sequential niche technique for multimodal function optimization. Evolutionary computation 1, 101-125.

Bojarczuk, C. C., Lopes, H. S., Freitas, A. A., 2000. Genetic programming for knowledge discovery in chest-pain diagnosis. IEEE Eng Med Biol Mag 19, 38-44.

Bojarczuk, C. C., Lopes, H. S., Freitas, A. A., Michalkiewicz, E. L., 2004. A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets. Artif Intell Med 30, 27-48.

Booker, L. B., 1982. Intelligent Behavior as an Adaptation to the Task of Environment. University of Michigan.

Brameier, M., Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. Transactions on Evolutionary Computation, IEEE 5, 17-26.

Briggs, F. B., Ramsay, P. P., Madden, E., Norris, J. M., Holers, V. M., Mikuls, T. R., Sokka, T., Seldin, M. F., Gregersen, P. K., Criswell, L. A., Barcellos, L. F., 2010. Supervised machine learning and logistic regression identifies novel epistatic risk factors with PTPN22 for rheumatoid arthritis. Genes Immun 11, 199-208.

Brindle, A., 1981. Genetic Algorithms for Function Optimization University of Alberta.

Brión, M., 2008. Estrategias con genes candidatos: búsqueda de SNPs. Jornadas de formación CeGen "Estudios de asociación en cáncer: Desde genes candidatos a GWAs", Santiago de Compostela.

Britannica, 2014. Single Nucleotide Polymorphism (SNP). Encyclopædia Britannica. http://www.britannica.com/EBchecked/topic/1334681/single-nucleotide-polymorphism.

Butler, J. M., Tsang, E. P. K., 1995. EDDIE beats the bookies. Technical Report CSM-259. Computer Science, University of Essex, Colchester, UK.

Canavan, F., Harding, S., Gustard, L., Murphy, A. M., Miller, J. F., Smith, S. L., 2012. Computer-aided detection of screening breast cancer: a novel approach based on genetic programming. Breast Cancer Research 14.

Cardno, A. G., Gottesman, II, 2000. Twin studies of schizophrenia: from bow-and-arrow concordances to star wars Mx and functional genomics. Am J Med Genet 97, 12-7.

Clark, T. G., De Iorio, M., Griffths, R. C., 2008. An evolutionary algorithm to find associations in dense genetic maps. IEEE Transactions on Evolutionary Computation 12, 297-306.

Clark, T. G., De Iorio, M., Griffiths, R. C., Farrall, M., 2005. Finding associations in dense genetic maps: a genetic algorithm approach. Hum Hered 60, 97-108.

Cohen, W. W., 1995. Fast Effective Rule Induction. Twelfth International Conference on Machine Learning, pp. 115-123.

Cook, J. W., Cunningham, W. H., Pulleyblank, W. R., Schrijver, A., 1997. Combinatorial Optimization. John Wiley & Sons, New York.

Cook, N. R., Zee, R. Y., Ridker, P. M., 2004. Tree and spline based association analysis of gene-gene interaction models for ischemic stroke. Stat Med 23, 1439-53.

Cordell, H. J., Clayton, D. G., 2002. A unified stepwise regression procedure for evaluating the relative effects of polymorphisms within a gene using case/control or family data: application to HLA in type 1 diabetes. Am J Hum Genet 70, 124-41.

Cramer, M. L., 1985. A Representation for the Adaptive Generation of Simple Sequential Programs. Proceedings of an International Conference on Genetic Algorithms and their Applications, Erlbaum.

Chamberlain, C., 2008. Uso de biomarcadores y validación en el desarrollo clínico de fármacos. . Biomarcadores y medicina traslacional, UIMP.

Chellapilla, K., 1997. Evolutionary programming with tree mutations: Evolving computer programs without crossover. Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan Kaufmann, Stanford University, CA, pp. 431-438.

Chinchilla Moreno, A., 2007. Las esquizofrenias. Sus hechos y valores clínicos y terapéuticos. Elsevier Masson.

Dai, S. P., Gao, L., Zhu, Q., Zhu, C., 2007. A novel genetic algorithm based on image databases for mining association rules. In: Lee, R., et al., Eds.), 6th IEEE/ACIS International Conference on Computer and Information Science, pp. 977-980.

Darwin, C., 1859. On the Origin of Species by Means of Natural Selection. John Murray, London.

De Jong, A. K., 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems University of Michigan.

De Jong, K. A., 1988. Learning with genetic algorithms: an overview. Machine Learning 3, 121-138.

De Jong, K. A., Spears, W. M., Gordon, D. F., 1993. Using genetic algorithms for concept learning. Machine Learning 13, 161-188.

De Vries, H., 1889. Befruchtung und Bastardirung. Intracellular Pangenesis. Including a paper on Fertilization and Hybridization. Translated from the German by C. Stuart Gager in 1910. The Open Court Publishing Co., Chicago.

Delen, D., Walker, G., Kadam, A., 2005. Predicting breast cancer survivability: a comparison of three data mining methods. Artif Intell Med 34, 113-27.

den Dunnen, J. T., Antonarakis, S. E., 2000. Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion. Hum Mutat 15, 7-12.

Deshpande, M., Karypis, G., 2002. Using Conjunction of Attribute Values for Classification. XI CIKM. ACM Press, pp. 356-364.

Engoren, M., Habib, R. H., Dooner, J. J., Schwann, T. A., 2013. Use of genetic programming, logistic regression, and artificial neural nets to predict readmission after coronary artery bypass surgery. J Clin Monit Comput 27, 455-64.

Farinaccio, A., Vanneschi, L., Giacobini, M., Mauri, G., Provero, P., 2010. On the use of genetic programming for the prediction of survival in cancer. Proceedings of the

12th annual conference on Genetic and evolutionary computation. ACM, pp. 163-170.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. From Data Mining to Knowledge Discovery in Databases. AI Magazine. AAAI.

Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C., 1991. Knowledge Discovery In Databases: An Overview. In: Piatetsky-Shapiro, G., Frawley, W. J., Eds.), Knowledge Discovery In Databases. AAAI Press/MIT Press, Cambridge, MA, pp. 1-30.

Freitas, A. A., 1997. A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction. Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan-Kauffman, Stanford University, CA.

Freund, Y., Mason, L., 1999. The alternating decision tree learning algorithm. ICML, pp. 124-133.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics 28, 337-407.

Fuchs, M., 1999. Large Populations Are Not Always The Best Choice In Genetic Programming. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99). Morgan-Kauffman, Orlando, FL, pp. 1033-1038.

Fujiki, C., Dickinson, J., 1987. Using the Genetic Algorithm to Generate LISP Source Code to Solve the Prisoner's Dilemma. Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Erlbaum.

Gaines, B. R., Compton, P., 1995. Induction of Ripple-Down Rules Applied to Modeling Large Databases. Journal of Intelligent Information Systems 5, 221-228.

Gathercole, C., Ross, P., 1997. Small Populations over Many Generations can beat Large Populations over Few Generations in Genetic Programming. . In: Koza, J. R., et al., Eds.), Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan Kaufmann, Stanford University, CA, pp. 111-118.

Giordana, A., Neri, F., 1995. Search intensive concept induction. Evolutionary computation 3, 375-416.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Boston, MA.

Greene, D. P., F., S. S., 1993. Competition based induction of decision models from examples. Machine Learning 13, 229-257.

Gruau, F., 1992. Genetic synthesis of boolean neural networks with a cell rewriting developmental process. In: Whitley, D., Schafer, J. D., Eds.), Proceedings of the Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN92). IEEE Computer Soc., Los Alamitos, CA, pp. 55-74.

Gruau, F., 1994. Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm. Laboratoire de l'Informatique du Parallilisme. Ecole Normale Supirieure de Lyon, France.

Guo, H., Nandi, A. K., 2006. Breast cancer diagnosis using genetic programming generated feature. Pattern Recognition 39, 980-987.

Hall, M., Frank, E., 2008. Combining Naive Bayes and Decision Tables. Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), Florida.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., 2009. The WEKA Data Mining Software: An Update. SIGKDD Explorations 11.

HapMap, 2014. International HapMap Project. http://www.hapmap.org/.

Hernández, J., Ramírez, M. J., Ferri, C., 2004. Introducción a la Minería de datos. Pearson, Prentice Hall.

Hipp, J., Guntzer, U., Nakaeizadeh, G., 2000. Algorithms for Association Rule Mining - a General Survey and Comparison. In: Fayyad, U., (Ed.), Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 2. ACM, Boston, pp. 58 - 64.

Hoffmann, F., 2004. Combining boosting and evolutionary algorithms for learning of fuzzy classification rules. Fuzzy Sets and Systems 141, 47-58.

Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., Hall, M., 2001. Multiclass alternating decision trees. ECML, pp. 161-172.

Holland, J. H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Michigan.

Holland, J. H., Reitman, J. S., 1978. Cognitive Systems Based on Adaptative Algorithms. In: Waterman, D. A., Hayes-Roth, F., Eds.), Pattern-Directed Inference Systems. Academic Press, New York, NY, pp. 313-329.

Hong, J. H., Cho, S. B., 2004. Lymphoma cancer classification using genetic programming with SNR features. In: Keijzer, M., et al., Eds.), Genetic Programming, Proceedings, Vol. 3003, pp. 78-88.

Hubley, R. M., Zitzler, E., Roach, J. C., 2003. Evolutionary algorithms for the selection of single nucleotide polymorphisms. BMC Bioinformatics 4, 30.

Ingraham, L. J., Kety, S. S., 2000. Adoption studies of schizophrenia. Am J Med Genet 97, 18-22.

Janikow, C. Z., 1993. A knowledge-intensive genetic algorithm for supervised learning. Machine learning 13, 189-228.

Johanson, B., 1997. Automated fitness raters for GP-music system. School of Computer Science, Vol. MsC. University of Birmingham, Birmingham, UK.

Johanson, B., Poli, R., 1998. GP-music: An interactive genetic programming system for music generation with automated fitness raters. Technical Report CSRP-98-13. University of Birmingham, School of Computer Science.

Jourdan, L., Dhaenens, C., Talbi, E.-G., Gallina, S., 2002. A Data Mining Approach to Discover Genetic and Environmental Factors involved in Multifactorial Diseases. Knowledge Based Systems 15, 235-242.

Kelemen, A., Vasilakos, A. V., Liang, Y., 2009. Computational Intelligence for genetic association study in complex diseases: review of theory and applications.

International Journal of Computational Intelligence in Bioinformatics and System Biology 1, 15-31.

Kendler, K. S., Diehl, S. R., 1993. The genetics of schizophrenia: a current, genetic-epidemiologic perspective. Schizophrenia Bulletin 19, 261-285.

Kohavi, R., 1995a. The Power of Decision Tables. 8th European Conference on Machine Learning. Springer-Verlag, Heraclion, Greece, pp. 174-189.

Kohavi, R., 1995b. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14th International Joint Conference on Artificial Intelligence, Vol. 2. Morgan Kaufmann, Montreal, Quebec, Canada, pp. 1137-1143.

Kohavi, R., 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. KDD, pp. 202-207.

Koza, J. R., 1990. Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. Technical Report No. STAN-CS-90-314. Computer Sciences Department, Stanford University.

Koza, J. R., 1992. Genetic Programming. On the Programming of Computers by means of Natural Selection. The MIT Press, Cambrige, MA.

Kreiner, T., Tillman Buck, K., 2005. Moving toward whole-genome analysis: A technology perspective. American Journal of Health-System Pharmacy 62, 296-305.

Kuo, W. J., Chang, R. F., Chen, D. R., Lee, C. C., 2001. Data mining with decision trees for diagnosis of breast tumor in medical ultrasonic images. Breast Cancer Res Treat 66, 51-7.

Langdon, W., Graillet, O. S., Harrison, A., 2010. Automated DNA Motif Discovery. arXiv:1002.0065.

Langdon, W. B., Buxton, B. F., 2004. Genetic programming for mining DNA chip data from cancer patients. Genetic Programming and Evolvable Machines 5, 251-257.

Li, S. X., Chen, Q. Y., Zhang, Y. J., Liu, Z. M., Xiong, H. L., Guo, Z. Y., Mai, H. Q., Liu, S. H., 2012. Detection of nasopharyngeal cancer using confocal Raman spectroscopy and genetic algorithm technique. Journal of Biomedical Optics 17.

Li, W., Han, J., Pei, J., 2001. CMAR: Accurate and Efficcient Classification Based on Multiple Class-Association (ICDM). IEE International Conference on Data Mining, pp. 169-376.

Little, R. J. A., Rubin, D. B., 2002. Statistical Analysis with Missing Data. John Wiley, New York.

Liu, B., Hsu, W., Ma, Y., 1998. Integrating Classification and Association Rule Mining. Fourth International Conference on Knowledge Discovery and Data Mining (KDD), New York, USA, pp. 80-86.

Liu, B., Ma, Y., Wong, C. K., 2001. Classification Using Association Rules: Weaknesses and Enhancements. In: Kumar, V., (Ed.), Data mining for scientific applications.

Lucrédio, D., Almeida, E. S., Prado, A. F., 2004. A Survey on Software Components Search and Retrieval. In: Steinmetz, R., Mauthe, A., Eds.), 30th IEEE EUROMICRO Conference. Component-Based Software Engineering Track. IEEE Press, Rennes, pp. 152-159.

Luke, S., 2000. Two Fast Tree-Creation Algorithms for Genetic Programming. IEEE Transactions on Evolutionary Computation.

Martin, B., 1995. Instance-Based learning: Nearest Neighbor With Generalization. In: University of Waikato, D. o. C. S., (Ed.), Computer Science Working Papers, Hamilton, New Zealand.

Mata, J., Alvarez, J. L., Riquelme, J. C., 2001. Mining Numeric Association Rules with Genetic Algorithms. In: Kurkova, V., et al., Eds.), 5th Internacional Conference on Artificial Neural Networks and Genetic Algorithms, Vol. Artificial Neural Nets and Genetic Algorithms. Springer Computer Science, Praga, pp. 264-267.

McGue, M., Gottesman, II, 1989. A single dominant gene still cannot account for the transmission of schizophrenia. Arch Gen Psychiatry 46, 478-80.

McGuffin, P., Owen, M. J., Farmer, A. E., 1995. Genetic basis of schizophrenia. Lancet 346, 678-82.

McLachlan, G. J., Do, K.-A., Ambroise, C., 2004. Analyzing Microarray Gene Expression Data. Wiley-Interscience, Hoboken, New Jersey.

Mendel, J. G., 1865. Versuche über Plflanzenhybriden Verhandlungen des naturforschenden Vereines in Brünn. Bd. IV für das Jahr. Abhandlungen, pp. 3-47.

Mitra, A. P., Almal, A. A., George, B., Fry, D. W., Lenehan, P. F., Pagliarulo, V., Cote, R. J., Datar, R. H., Worzel, W. P., 2006. The use of genetic programming in the analysis of quantitative gene expression profiles for identification of nodal status in bladder cancer. BMC Cancer 6, 159.

Molitor, J., Marjoram, P., Thomas, D., 2003. Fine-scale mapping of disease genes with multiple mutations via spatial clustering techniques. Am J Hum Genet 73, 1368-84.

Montana, D. J., 1995. Strongly typed genetic programming. Evolutionary Computation 3, 199-200.

Mooney, M. A., Wilmot, B., McWeeney, S. K., Bipolar Genome, S., 2012. The GA and the GWAS: Using Genetic Algorithms to Search for Multilocus Associations. Ieee-Acm Transactions on Computational Biology and Bioinformatics 9, 899-910.

Moore, J. H., 2004. Computational analysis of gene-gene interactions using multifactor dimensionality reduction. Expert Rev Mol Diagn 4, 795-803.

Moore, J. H., White, B. C., 2006. Exploiting expert knowledge for genome-wide genetic analysis using genetic programming. In: Runarsson, T. P., et al., Eds.), Parallel Problem Solving from Nature – PPSN IX, Vol. 4193. Springer, Reykjavik, pp. 969-977.

Motsinger, A. A., Lee, S. L., Mellick, G., Ritchie, M. D., 2006. GPNN: power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. BMC Bioinformatics 7, 39.

Motulsky, A. G., 2006. Genetics of complex diseases. J Zhejiang Univ Sci B 7, 167-8.

Nägeli, C., 1842. Zur Entwickelungsgeschichte des Pollens bei den Phanerogamen. Ovell & Füssli, Zürich.

NHGRI, 2014. National Human Genome Research Institute. Glosario de términos genéticos. Herencia mendeliana. http://www.genome.gov/GlossaryS/index.cfm?id=122.

Oakley, H., 1993. Signal filtering and data processing for laser rheometry. Technical report. Institute of Naval Medicine, Portsmouth, UK.

Oakley, H., 1994a. Two scientific applications of genetic programming: Stack filters and non-linear equation fitting to chaotic data. Advances in Genetic Programming. MIT Press, pp. 369-389.

Oakley, H., 1994b. The application of genetic programming to the investigation of short, noisy, chaotic data series. Lecture Notes in Computer Science Proceedings of the Workshop on Artificial Intelligence and Simulation of Behaviour Workshop on Evolutionary Computing.

Paul, T. K., Iba, H., 2009. Prediction of cancer class with majority voting genetic programming classifier using gene expression data. IEEE/ACM Trans Comput Biol Bioinform 6, 353-67.

Pearl, J., 1984. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley.

Pereira, F. B., Machado, P., Costa, E., Cardoso, A., 1999. Graph based crossover-A case study with the busy beaver problem. Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99. Morgan Kaufmann, Orlando, FL, pp. 1149-1155.

Perry, J. E., 1994. The effect of population enrichment in genetic programming. Proceedings of the 1994 IEEE World Congress on Computational Intelligence. IEEE Press, Orlando, FL, pp. 456-461.

Picard, R., Cook, D., 1984. Cross-Validation of Regression Models. Journal of the American Statistical Association 79, 575–583.

Poli, R., Langdon, W. B., 1998. On the ability to search the space of programs of standard, one-point and uniform crossover in genetic programming. Technical Report CSRP-98-7. University of Birmingham, School of Computer Science.

Qodmanan, H. R., Nasiri, M., Minaei-Bidgoli, B., 2011. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. Expert Systems with Applications 38, 288-298.

Quinlan, J. R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

Randall, M. C., Thorne, C. E., Wild, C. A., 1994. Standard Comparison of Adaptive Controllers to Solve the Cart Pole Problem. Proceedings of the Second IEEE Australian and New Zealand Conference on Intelligent Information Systems, pp. 61-65.

Reif, D. M., White, B. C., Moore, J. H., 2004. Integrated analysis of genetic, genomic and proteomic data. Expert Rev Proteomics 1, 67-75.

Reynolds, C. W., 1992. An evolved, vision-based behavioral model of coordinated group motion. From Animals to Animals (Proceedings of Simulation of Adaptive Behaviour). MIT Press.

Reynolds, C. W., 1994. Evolution of obstacle avoidance behaviour: using noise to promote robust solutions. Advances in Genetic Programming. MIT Press, pp. 221-241.

Ritchie, M. D., White, B. C., Parker, J. S., Hahn, L. W., Moore, J. H., 2003. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. BMC Bioinformatics 4, 28.

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., Moore, J. H., 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet 69, 138-47.

Rivero, D., Rabuñal, J. R., Dorado, J., Pazos, A., 2004. Using Genetic Programming for Character Discrimination in Damaged Documents. Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC, pp. 349-358.

Rivero, D., Rabuñal, J., Dorado, J., Pazos, A., 2005. Time series forecast with anticipation using genetic programming. 8th International Work-conference on Artificial Neural Networks. Springer, pp. 968–975.

Rova, M., Haataja, R., Marttila, R., Ollikainen, V., Tammela, O., Hallman, M., 2004. Data mining and multiparameter analysis of lung surfactant protein genes in bronchopulmonary dysplasia. Hum Mol Genet 13, 1095-104.

Roy, S., 2002. Nearest Neighbor With Generalization. University of Canterbury, Christchurch, New Zealand.

Sáiz, J., Fañanás, L., 1998. Introducción: Genética y Psiquiatría. Monografías de Psiquiatría 10.

Sáiz Ruiz, J., 1999. Esquizofrenia: Enfermedad del cerebro y reto social. Masson, Barcelona.

Segovia, J., Isasi, P., 1997. Genetic Programming For Designing Ad Hoc Neural Network Learning Rules. Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan-Kauffman, Stanford University, CA.

Sham, P., 1996. Genetic epidemiology. Br Med Bull 52, 408-33.

Smith, S. F., 1980. A learning system based on genetic adaptive Algorithms. Vol. Ph.D. University of Pittsburgh.

Soule, T., 1998. Code Growth in Genetic Programming. University of Idaho.

Soule, T., Foster, J. A., 1997. Code Size and Depth Flows in Genetic Programming. Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan Kauffmann, San Francisco, CA, pp. 313-320.

Spencer, G. F., 1994. Automatic generation of programs for crawling and walking. Advances in Genetic Programming. MIT Press, pp. 335-353.

Srikant, R., Agrawal, R., 1995. Mining generalized association rules. In: Dayal, U., et al., Eds.), 21st International Conference on Very Large Data Bases. Morgan Kaufmann, Zurich, pp. 407-419.

Tackett, W. A., Carmi, A., 1994. The donut problem: Scalability and generalization in genetic programming. Advances in Genetic Programming. MIT Press, pp. 143-176.

Tahri-Daizadeh, N., Tregouet, D. A., Nicaud, V., Manuel, N., Cambien, F., Tiret, L., 2003. Automated detection of informative combined effects in genetic association studies of complex traits. Genome Res 13, 1952-60.

Tan, P., Steinbach, M., Kumar, V., 2006. Introduction to Data Mining. Pearson, Addison-Wesley.

Thomas, D. C., 2004. Statistical Methods in Genetic Epidemiology. Oxford University Press, Ofxord.

Toivonen, H., Onkamo, P., Hintsanen, P., al., e., 2005. Data mining for gene mapping. In: Kantardzic, M. M., Zurada, J., Eds.), New Generation of Data Mining Applications. IEEE Press, Hobokonen, NJ, pp. 263-293.

Tzeng, J. Y., Devlin, B., Wasserman, L., Roeder, K., 2003. On the identification of disease mutations by the analysis of haplotype similarity and goodness of fit. Am J Hum Genet 72, 891-902.

Uhmn, S., Kim, D.-H., Ko, Y.-W., Cho, S., Cheong, J., Kim, J., 2009. A study on application of single nucleotide polymorphism and machine learning techniques to diagnosis of chronic hepatitis. Expert Systems 26, 60-69.

Venturini, G., 1993. SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts. In: Brazdil, P., (Ed.), European Conference on Machine Learning (ECML), Vol. 667. Springer, Vienna, pp. 280-296.

Vermeulen-Jourdan, L., Dhaenens, C., Talbi, E.-G., 2005. Linkage disequilibrium study with a parallel adaptive GA. International Journal of Foundation in Computer Science 16, 241-260.

Waddell, M., Page, D., Zhan, F., al., e., 2005. Predicting cancer susceptibility from single-nucleotide polymorphism data: A case study in multiple myeloma. In: Parthasarathy, S., et al., Eds.), Fifth ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD). ACM, Chicago, pp. 21-28.

Wang, H., Zhang, Y. M., Li, X., Masinde, G. L., Mohan, S., Baylink, D. J., Xu, S., 2005. Bayesian shrinkage estimation of quantitative trait loci parameters. Genetics 170, 465-80.

Watson, J. D., Crick, F. H., 1953. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. Nature 171, 737-8.

Wetzel, A., 1983. Evaluation of the Effectiveness of Genetic Algorithms in Combinational Optimization University of Pittsburg.

Wilcox, M. A., Wyszynski, D. F., Panhuysen, C. I., Ma, Q., Yip, A., Farrell, J., Farrer, L. A., 2003. Empirically derived phenotypic subgroups - qualitative and quantitative trait analyses. BMC Genet 4 Suppl 1, S15.

Winkler, S. M., Affenzeller, M., Wagner, S., 2009. Using enhanced genetic programming techniques for evolving classifiers in the context of medical diagnosis. Genetic Programming and Evolvable Machines 10, 111-140.

Witten, I. H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann

Wittke-Thompson, J. K., Pluzhnikov, A., Cox, N. J., 2005. Rational inferences about departures from Hardy-Weinberg equilibrium. Am J Hum Genet 76, 967-86.

Yan, X. W., Zhang, C. Q., Zhang, S. C., 2009. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Systems with Applications 36, 3066-3076.

Yang, C. H., Chuang, L. Y., Chen, Y. J., Tseng, H. F., Chang, H. W., 2011. Computational Analysis of Simulated SNP Interactions Between 26 Growth Factor-Related Genes in a Breast Cancer Association Study. Omics-a Journal of Integrative Biology 15, 399-407.

Young, S. S., Ge, N., 2005. Recursive partitioning analysis of complex disease pharmacogenetic studies. I. Motivation and overview. Pharmacogenomics 6, 65-75.

Glosario

Ácido desoxirribonucleico Constituye el material genético de las células y contiene en su secuencia la información para la síntesis de proteínas.

Algoritmo genético Esta técnica surgió como una aproximación de inteligencia artificial. Se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Alelo Cada uno de los genes del par que ocupa el mismo lugar en los cromosomas homólogos. Su expresión determina el mismo carácter o rasgo de organización, como el color de los ojos.

Computación evolutiva Rama de la inteligencia artificial que involucra problemas de optimización combinatoria. Se inspira en los mecanismos de la evolución biológica.

Cromosoma Filamento condensado de ácido desoxirribonucleico, visible en el núcleo de las células durante ciertas divisiones

	celulares. Su número es constante para cada especie animal o vegetal.
Enfermedad compleja	Enfermedad en la que establecer una relación gen-enfermedad resulta difícil.
Especificidad	Proporción de sanos correctamente identificados.
Estudio de asociación	Estudio que compara el ADN de personas con una enfermedad con el ADN de personas que no padecen la enfermedad.
Exposoma	Representa la variabilidad ambiental y se puede definir como la suma total de los múltiples factores a los que está expuesta una persona a lo largo de su vida: la dieta, el estilo de vida, el uso (y abuso) de fármacos, la contaminación, el contacto con productos químicos, las infecciones que sufrimos, el estrés y todos los factores ambientales internos y externos a los que se expone una persona desde que nace.
Fenotipo	Manifestación visible del genotipo en un determinado ambiente.
Gen	Secuencia de ADN que constituye la unidad funcional para la transmisión de los caracteres hereditarios.
Genotipo	Conjunto de los genes de un individuo, incluida su composición alélica.
Nucleótido	Compuesto orgánico constituido por una base nitrogenada, un azúcar y ácido fosfórico, que representa la

	unidad estructural del ADN.
Odds ratio	Posibilidad de que una condición de salud o enfermedad se presente en un grupo de población frente al riesgo de que ocurra en otro.
Optimización combinatoria	Es una rama de las matemáticas aplicadas y las ciencias de la computación que consiste en la búsqueda de la solución óptima a partir de un conjunto finito de objetos. Esto se lleva a cabo reduciendo el tamaño efectivo del espacio de búsqueda y explorándolo eficientemente.
Programación Genética	Técnica basada en los algoritmos evolutivos e inspirada en la evolución biológica para desarrollar automáticamente programas que realicen una tarea definida por el usuario. Es una especialización de los algoritmos genéticos. Es una técnica de aprendizaje automático utilizada para optimizar una población de programas de acuerdo a una función de ajuste o *fitness* que evalúa la capacidad de cada programa para llevar a cabo la tarea en cuestión.
Valor predictivo negativo	Probabilidad de no tener la enfermedad si el resultado de la prueba diagnóstica es negativo.
Valor predictivo positivo	Probabilidad de tener la enfermedad si el resultado de la prueba diagnóstica es positivo.
Sensibilidad	Proporción de enfermos correctamente identificados.
Single Nucleotide Polymorphism	Variación en la secuencia de ADN que sucede cuando un nucleótido difiere entre dos miembros de una especie o entre dos cromosomas de un par en un individuo. Para

que una variación sea considerada un SNP, debe aparecer en al menos 1% de la población.

Anexos

A continuación se incluyen los artículos de investigación originales publicados en revistas de impacto más relacionados con respecto al estudio computacional realizado en esta tesis, describiendo brevemente cada una de ellos.

1. Machine Learning Techniques for Single Nucleotide Polymorphism - Disease Classification Models in Schizophrenia

Este artículo fue el primero publicado sobre el trabajo realizado por la doctoranda, como parte de la línea de investigación que daría fruto a la presente tesis, en una revista de impacto (en 2010, Q2 con I.F.: 1.988).

El objetivo de dicho trabajo era la comparación de diversos métodos de clasificación, entre ellos una versión inicial de la *aproximación 1* presentada en esta tesis (referenciada como EC), para tratar de extraer patrones que relacionasen SNPs con la predisposición al desarrollo de la esquizofrenia. En base a estos modelos de clasificación y, de forma similar a la utilización de *Quantitative Structure - Activity Relationships* (QSARs) o *Quantitative Protein (or Proteome)-Disease Relationships* (QPDRs), ampliamente empleados para la predicción de propiedades de proteínas y de enfermedades respectivamente, se establece un nuevo modelo denominado *Quantitative Genotype – Disease Relationships* (QGDRs) que relaciona secuencias de moléculas de ácidos nucleicos con la esquizofrenia. Este modelo es capaz de reconocer secuencias de ADN de forma automática y clasificar correctamente entre un 78,3% y un 93,8% de sujetos, utilizando desde el conjunto de datos clínicos reales con un mínimo de controles simulados, así como conjuntos que contenían un mayor número de controles adicionales simulados utilizando HAP-SAMPLE.

Este estudio computacional se realizó empleando doce técnicas diferentes (métodos basados en redes de neuronas artificiales, redes bayesianas, máquinas de soporte vectorial, tablas de decisión…) para generar 252 modelos de clasificación, utilizando el total de SNPs contenidos en el fichero de datos original, así como analizando por separado los SNPs de los distintos genes a los que pertenecían (HTR2A y DRD3). El método que obtenía mejores resultados con el mínimo número de sujetos simulados estaba basado en redes de neuronas artificiales. Sin embargo, la *aproximación 2b* mostrada en esta tesis supera el porcentaje de clasificación obtenido por el mejor método aplicado en este estudio sobre el total de datos clínicos reales originales.

2. Applied Computational Techniques on Schizophrenia Using Genetic Mutations

Este artículo continúa en la misma línea que el anterior, presentando no sólo un estudio computacional, sino la implementación de uno de los modelos QDGR que obtenía mejores resultados mediante una herramienta accesible de forma gratuita on-line, además de combinar un método de selección de variables basado en redes de neuronas y computación evolutiva con un clasificador SVM. También ha sido publicado en una revista de impacto (aún no ha sido publicado el factor de impacto para 2013; pero en 2012, Q1 con I.F.: 3.702).

La herramienta que implementa el modelo QDGR (SNPSchizo), se enmarca dentro de un portal que ofrece diversos modelos teóricos basados en inteligencia artificial, biología computacional y bioinformática, aplicados al estudio de sistemas complejos en lo que se conoce como las "ÓMICAS" (genómica, transcriptómica, metabolómica…), en especial aquellos relevantes para el cáncer, la neurociencia o la microbiología, entre otros.

Es importante resaltar que el modelo que obtenía mejores resultados y que se implementó on-line utiliza 40 variables, lo cual es un número muy alto. A diferencia de este modelo, el método que combina selección de variables (basado en redes de neuronas y AA.GG.) con un clasificador (SVM), es capaz de obtener un modelo utilizando solamente 17 variables y alcanzando un porcentaje de clasificación del 78,2%.

Finalmente, es interesante mencionar que las variables obtenidas durante la fase de selección de variables incluyen SNPs que respaldan las conclusiones obtenidas por el grupo

liderado por Ángel Carracedo, además de algunos SNPs adicionales que se encuentran en la misma región si se considera un modelo que contiene 21 variables.

Molecules **2010**, *15*, 4875-4889; doi:10.3390/molecules15074875

OPEN ACCESS

molecules

ISSN 1420-3049

www.mdpi.com/journal/molecules

Article

Machine Learning Techniques for Single Nucleotide Polymorphism—Disease Classification Models in Schizophrenia

Vanessa Aguiar-Pulido, José A. Seoane, Juan R. Rabuñal, Julián Dorado, Alejandro Pazos and Cristian R. Munteanu *

Department of Information and Communication Technologies, Computer Science Faculty, University of A Coruña, Campus de Elviña, S/N, 15071 A Coruña, Spain; E-Mails: vanesa.aguiar@udc.es (V.A.-P.); jseoane@udc.es (J.A.S.); juanra@udc.es (J.R.R.); julian@udc.es (J.D.); apazos@udc.es (A.P.)

* Author to whom correspondence should be addressed; E-Mail: muntisa@gmail.com; Tel.: +34 981 167 000 Ext. 1302; Fax: +34 981 167 160.

Received: 4 June 2010; in revised form: 8 July 2010 / Accepted: 9 July 2010 / Published: 12 July 2010

Abstract: Single nucleotide polymorphisms (SNPs) can be used as inputs in disease computational studies such as pattern searching and classification models. Schizophrenia is an example of a complex disease with an important social impact. The multiple causes of this disease create the need of new genetic or proteomic patterns that can diagnose patients using biological information. This work presents a computational study of disease machine learning classification models using only single nucleotide polymorphisms at the HTR2A and DRD3 genes from Galician (Northwest Spain) schizophrenic patients. These classification models establish for the first time, to the best knowledge of the authors, a relationship between the sequence of the nucleic acid molecule and schizophrenia (Quantitative Genotype – Disease Relationships) that can automatically recognize schizophrenia DNA sequences and correctly classify between 78.3–93.8% of schizophrenia subjects when using datasets which include simulated negative subjects and a linear artificial neural network.

Keywords: DNA molecule; SNP; schizophrenia; artificial neural networks; evolutionary computation

---

1. Introduction

Disease computational studies use diverse types of data, such as the structure and physical/chemical properties of a protein and DNA/RNA molecules, blood proteome mass spectra, DNA microarray results, disease biomarkers and concentration of the metabolites in physiological liquids. Schizophrenia, which is a common disease, can be defined as a heterogeneous syndrome characterized by perturbations in language, perception, thinking, social relationships and will. There is not a set of symptoms which uniquely characterize the disease, and even though researchers have been looking for a unique cause of schizophrenia for years with no success, most of them have concluded that schizophrenia would be the consequence of several cumulative effects of certain risk factors (genetic and environmental) [1]. Several studies of families, twins and foster-children confirmed and have allowed quantification of the contribution of genetics to schizophrenia [2]. After this, molecular genetics techniques started to be used to identify the genes that caused the disease [3]. These genes are not the genes of schizophrenia themselves, but rather they may transmit a set of characteristics which would increase the risk of developing the disease.

One of the most studied genes in relation to schizophrenia susceptibility is DRD3. As well as HTR2A, it is considered to be an important target for several antipsychotic drugs [4,5]. HTR2A encodes one of the receptors for serotonin and DRD3 encodes one subtype of the five dopamine receptors, both neurotransmitters. More specifically, Dopamine 3 receptors (DRD3) are concentrated in limbic regions of the brain, which are associated with cognitive, emotional and endocrine functions. Thus, it may be particularly relevant to schizophrenia [6], as the DRD3 messenger RNA is predominantly expressed in the limbic system, a region thought to be dysfunctional in this disease [7,8].

Association studies involving these functional candidate genes have systematically focused on a limited set of Single Nucleotide Polymorphisms (SNPs), generally based on previously reported small contributions of these markers of risk of susceptibility to schizophrenia. More specifically, SNP T102C (rs6313) at HTR2A and SNP Ser9Gly (rs6280) at DRD3 have been extensively analyzed in several schizophrenia case-control studies [9]. A SNP [10] is a single nucleotide site where two (of four) different nucleotides occur in a high percentage (*i.e.*, at least 1 %) of the population.

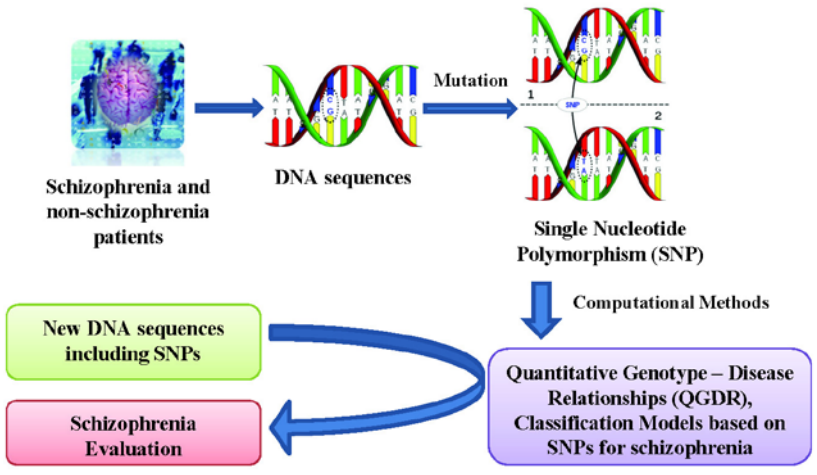
There are several studies on SNPs, such as that one in [11], where a method is presented for haplotype partitioning based on pairwise analysis of SNPs. A block-based approach for mapping a single locus trait was applied to blocks of different methods in a case-control study. Results show that any block-based association test is considerably more efficient than the conventional single site association trait and, in particular, the method presented performed best accuracy, even when a low marker density was available. Another study on SNPs is that one presented in [12]. In this paper, the use of two feature importance ranking measures (the modified t-test and F-statistics) is proposed to rank a large amount of SNPs and then the greedy manner together with a classifier are used in order to determine a desirable feature subset, which leads to the highest classification accuracy with the minimum size. Results show that both ranking methods are efficient at determining the important SNPs and they both find nearly the same amount of them. However, the first measure tends to be better in terms of classification accuracy. Compared to other methods, the results obtained in this paper are better.

There exist several genetic data simulation packages. Among those, we encounter coalescent-based methods [13], which have been used for population based simulation in genetic studies, such as GENOME [14]. This method was developed to overcome previous limitations. HAP-SAMPLE [15], which is the simulator used in this paper, uses the existing Phase I/II HapMap data to resample existing phased chromosomes to simulate datasets. There also exist forward-time population simulations, such as easyPOP [16], FPG [17], FREGENE [18], simuPOP [19] and genomeSIMLA [20]. The last method can simulate realistic patterns of LD in both family-based and case-control datasets and, unlike other similar packages, has proved to be an effective platform for simulating large scale genetic data. Another program capable of generating large scale genetic and also phenotypic variation data is presented in [21]. This program generates genotypes/phenotypes by perturbing real data, with the aim of creating a large number of replicates that share similar properties with real data.

Models based on Machine Learning have been extensively used to analyze complex diseases, such as diabetes [22], hepatitis [23], rheumatoid arthritis [24], *etc*. However, not many studies have been carried out on variation analysis in schizophrenia using Machine Learning algorithms [25]. Statistical models were the most used for this type of complex disease.

Quantitative Structure - Activity Relationships (QSARs) are widely used for predicting protein properties [26] and Quantitative Protein (or Proteome)-Disease Relationships (QPDRs) [27-33] for disease prediction. Recent works using complex networks of proteins or mass spectra of the human serum proteome have contributed to create theoretical models for cancer diagnosis and screening for cancer-related molecules in the case of colorectal [34,35], breast [34,36] and prostate [37-39] cancers. In a similar way, a Quantitative Genotype - Disease Relationship (QGDR) can be established in order to automatically evaluate schizophrenia DNA sequences using SNP data. Methods such as artificial neural networks [40], support vector machines [41], evolutionary computation [42,43] and other Machine Learning techniques [44] have been used in order to find the best classification models. This work presents a study of schizophrenia QGDR classification using only single nucleotide polymorphisms from Galician patients [9]. Thus, this information of the DNA molecule will be used as the input for several machine learning techniques that search for the best classification model capable of evaluating new schizophrenia DNA sequences (see Figure 1).

Figure 1. Flow chart of the QGDR model classification between the DNA structure (SNPs) and schizophrenia.



2. Results and Discussion

Two hundred and fifty two (252) QGDR classification models have been obtained using SNPs at two schizophrenia-related genes (each of them or both), twelve machine learning techniques and seven datasets, starting from the original data and using extra simulated negative (control) subjects (see Table 1). In terms of classification the subjects are organized in two groups: *Schizo* and *non-Schizo*. These models describe relationships between the DNA information (SNPs) and schizophrenia.

Table 1. The classification models obtained for the evaluated schizophrenia patients using the SNP information at DRD3 and HTR2A.

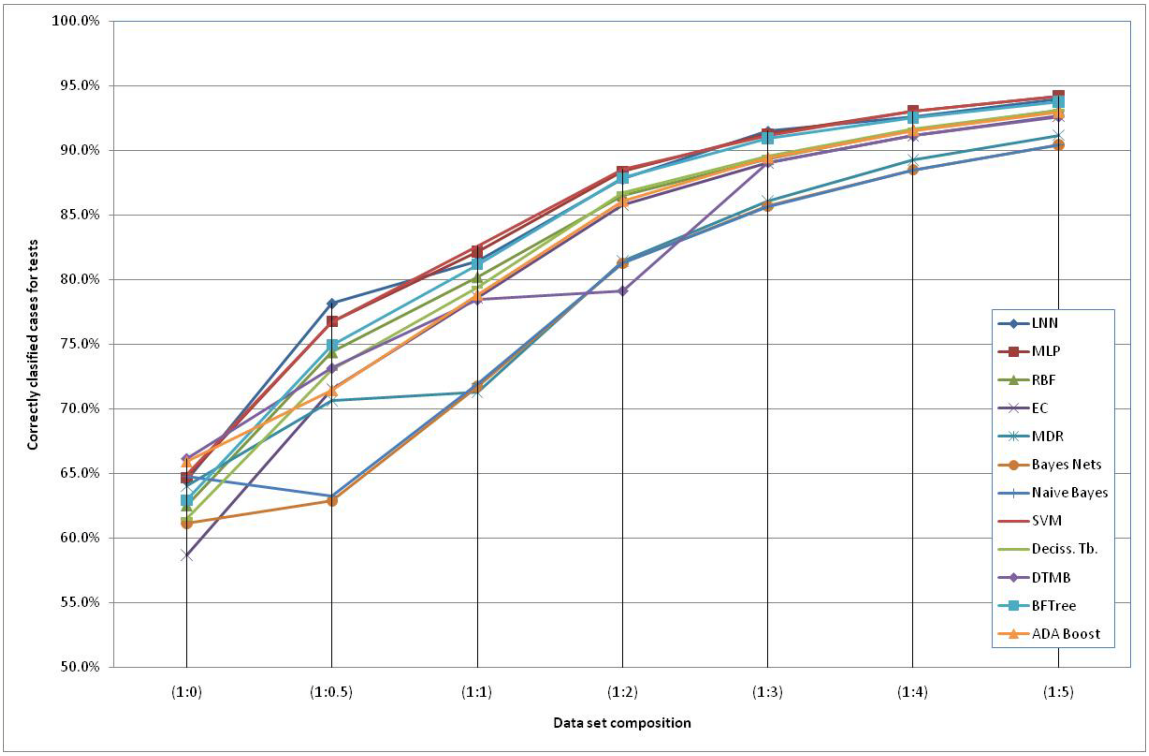
Data set	Gene	LNN	MLP	RBF	EC	MDR	Bayes Nets	Naïve Bayes	SVM	Decis. Tb.	DTNB	BFTree	AdaBoost
SNP (1:0)	**DRD3**	62.9%	59.5%	58.9%	56.6%	60.0%	62.5%	61.6%	64.8%	62.2%	59.5%	61.3%	63.4%
	HTR2A	62.4%	62.9%	63.7%	57.5%	64.0%	61.9%	66.6%	65.2%	61.0%	62.3%	62.8%	63.5%
	Both	64.5%	64.7%	62.5%	58.7%	64.0%	61.2%	64.8%	64.9%	61.5%	66.2%	62.9%	65.9%
SNP (1:0.5)	**DRD3**	74.6%	72.9%	71.5%	71.0%	60.5%	71.3%	71.0%	75.4%	73.5%	70.4%	73.7%	71.3%
	HTR2A	75.9%	75.5%	73.6%	71.7%	74.2%	62.2%	62.9%	77.4%	73.2%	70.9%	74.5%	71.4%
	Both	78.2%	76.8%	74.4%	71.5%	70.7%	62.9%	63.3%	76.8%	73.1%	73.2%	75.0%	71.4%
SNP (1:1)	**DRD3**	80.5%	79.5%	78.5%	78.2%	69.8%	77.9%	76.2%	81.4%	79.6%	77.1%	79.4%	78.6%
	HTR2A	80.7%	81.7%	80.2%	78.5%	71.0%	71.9%	72.3%	83.0%	79.8%	76.8%	81.2%	78.8%
	Both	81.4%	82.2%	80.2%	78.6%	71.3%	71.7%	72.0%	82.6%	79.4%	78.5%	81.2%	78.8%
SNP (1:2)	**DRD3**	87.0%	86.1%	85.8%	85.4%	79.4%	84.8%	83.2%	87.7%	86.6%	80.4%	86.1%	85.2%
	HTR2A	88.0%	88.1%	86.3%	85.9%	81.4%	81.3%	81.6%	88.8%	86.5%	76.2%	87.6%	86.1%
	Both	87.8%	88.4%	86.5%	85.8%	81.4%	81.3%	81.3%	88.5%	86.7%	79.2%	87.9%	86.1%
SNP (1:3)	**DRD3**	89.9%	89.5%	88.9%	88.4%	84.8%	89.4%	86.9%	90.6%	89.5%	87.6%	89.5%	88.7%
	HTR2A	90.4%	90.7%	89.3%	89.1%	85.9%	85.7%	85.9%	91.4%	89.7%	86.5%	90.3%	89.4%
	Both	91.5%	91.3%	89.3%	89.1%	86.1%	85.7%	85.6%	91.2%	89.5%	89.1%	90.9%	89.4%
SNP (1:4)	**DRD3**	91.9%	91.7%	91.3%	90.9%	87.4%	91.5%	89.2%	92.5%	91.6%	90.3%	91.5%	90.7%
	HTR2A	92.6%	92.7%	91.8%	91.2%	88.5%	88.6%	88.6%	93.2%	91.7%	88.5%	92.4%	91.5%
	Both	92.6%	93.0%	91.6%	91.2%	89.3%	88.5%	88.5%	93.0%	91.6%	91.1%	92.5%	91.5%
SNP (1:5)	**DRD3**	93.9%	93.1%	93.0%	92.1%	88.4%	92.9%	90.8%	93.6%	93.1%	91.8%	92.9%	92.2%
	HTR2A	93.2%	93.9%	92.9%	92.6%	91.2%	90.5%	90.5%	94.3%	93.1%	90.0%	93.5%	92.9%
	Both	93.9%	94.2%	93.1%	92.6%	91.2%	90.4%	90.4%	94.2%	93.1%	92.6%	93.8%	92.9%

Notes: LNN = Linear Neural Networks, MLP = Multilayer Perceptron; RBF = Radial Base Functions; EC = Evolutionary Computation; MDR = Multifactor Dimensionality Reduction; Bayes Nets = Bayesian Networks; SVM = Support Machine Vectors; Decis. Tb. = Decision Tables; DTNB = Decision Table Naïve Bayes Hybrid Classifier; BFTree = Best-First decision Tree classifier; AdaBoost = Adaptative Boosting.

The models generated using the original dataset correctly classify only 66.6% of the schizophrenic subjects when using the HTR2A gene and the Naïve Bayes method. This low accuracy can be due to the reduced number of subjects available and an increased number of "3" values of the SNPs (unknown data). Therefore, we included additional simulated subjects obtained with the HAP-SAMPLE software [15] in the negative group (*non-Schizo*), maintaining the capacity to evaluate positive subjects (cases) for the models. Thus, seven datasets have been created, labeled as SNP (1:*n*),

where 1:*n* (*n* = 0, 0.5, 1, 2, 3, 4, 5) is the proportion between the real subjects (positive and negative) and the simulated negative subjects (see details in the Experimental and Theoretical Section). The graphical representation of the evolution of the best classification depending on the additional number of simulated negative subjects is shown in Figure 2. It can observed that the classification percentages do not increase significantly after adding five parts of simulated negative subjects. Among the best models, we propose the following two QGDR models which correspond to simple linear artificial neural networks (LNN).

Figure 2. Correctly classified subjects depending on the simulated negative data for both genes; the dataset labels represent the proportion between real subjects (positive and negative = case and control) and simulated negative subjects.



The first model (Model 1) includes only a minimum number of simulated negative subjects, SNP (1:0.5): 260 real positive subjects, 354 real negative subjects and 307 simulated negative subjects for schizophrenia, a total of 921 subjects. It is based on 40 SNPs (at DRD3: rs7631540, rs6808291, rs1486012, rs9824856, rs2134655, rs963468, rs3773678, rs167771, rs226082, rs1486009, rs6280, rs7638876, rs9825563, rs1354348; at HTR2A: rs3889066, rs7329640, rs10507544, rs7333412, rs3125, rs6314, rs6308, rs1058576, rs1923884, rs2296972, rs9316233, rs659734, rs1928042, rs2770296, rs582385, rs1928040, rs731779, rs985934, rs9534505, rs6304, rs6305, rs2070036, rs6313, rs1328685, rs731244, rs10507547) and the model used was a LNN with 40 inputs and 152 neurons, which correctly classifies 78.2% of the subjects of the test group. The area under the receiver operating characteristic curve (AUC-ROC) for the cross-validation group (0.8405) shows that the model is not random (see Figure 3).

The second model (Model 2) includes a maximum number of simulated negative subjects, SNP (1:5): 260 real positive subjects, 354 real negative subjects and 3070 simulated negative subjects for

schizophrenia, a total of 3,684 subjects. The model is based only on two SNPs (rs7329640 and rs985934) at HTR2A: a LNN with two inputs and eight neurons, which correctly classifies 93.2% of the subjects of the test group. The AUC-ROC for the cross-validation group (0.9439) demonstrates the goodness of the model (see Figure 4).

Figure 3. Area under the receiver operating characteristic curve (AUC-ROC) for LNN 40:152-1:1 (Model 1).

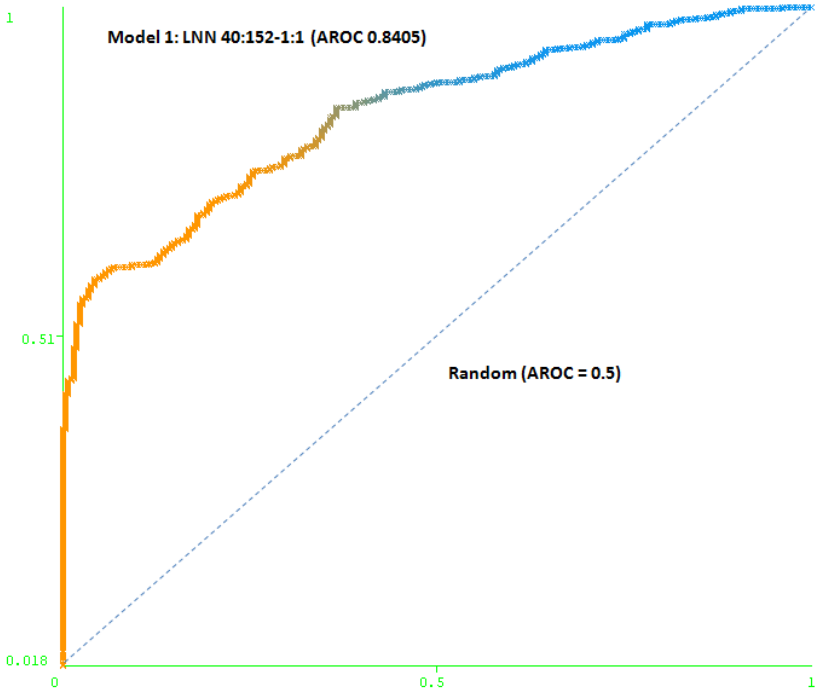


Figure 4. Area under the receiver operating characteristic curve (AUC-ROC) for LNN 2:8-1:1 (Model 2).

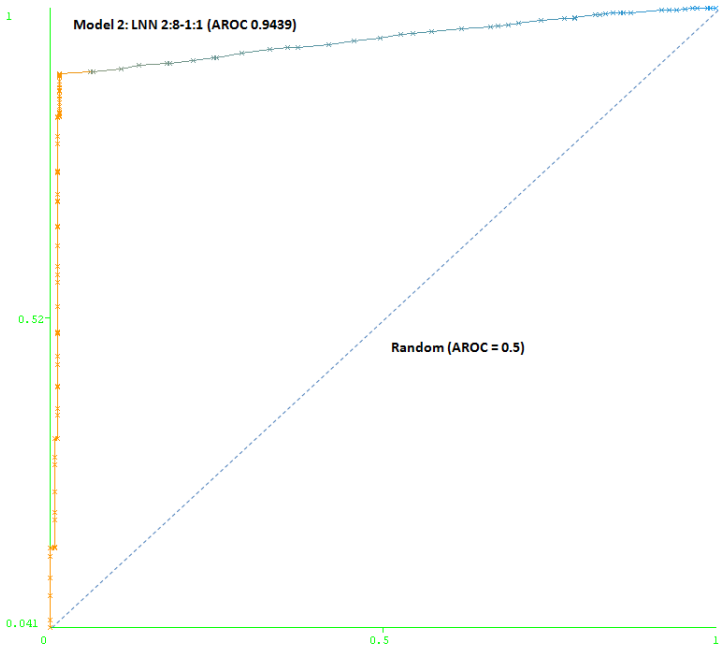


Table 1 shows that the classification accuracy percentages are in the range of 56.6–66.6% for SNP(1:0), 60.5–78.2% for SNP(1:0.5), 69.8–83.0% for SNP(1:1), 76.2–88.8% for SNP(1:2), 84.8–91.5% for SNP(1:3), 87.4–93.2% SNP(1:4) and 88.4–94.3% for SNP(1:5). In general, we can observe that the genotype information from the HTR2A gene is classifying more accurately than when considering the SNPs at DRD3 and using the Support Machine Vectors (SVM) technique [45]. There are two exceptions to this performance, with small differences, in the schizophrenia classification for SNP (1:0.5) and for SNP (1:3), where the maximum accuracy percentages correspond to LNN using information from both genes. Despite the fact that an MLP is more complex than an LNN, the first one obtains almost the same classification scores as the LNN. Finally, Evolutionary Computation (EC) [46] obtains better classification scores when the second gene or both genes together are considered, as a higher number of SNPs is taken into account and, thus, there is more information.

3. Experimental and Theoretical Section

3.1. Subjects and Genotyping

The case-control subjects consisted of 260 unrelated patients (65% males) being treated by the Galician Mental Health Service for schizophrenia and 354 unrelated blood negative donors (45% males) recruited from the Galician Blood Transfusion Centre (staff at the University of Santiago de Compostela and patients attending the University of Santiago de Compostela Hospital Complex). The study protocol was approved by the Bioethics Committee of the University of Santiago de Compostela (for details see [9]). In order to extract genomic DNA from white blood cells of peripheral venous blood from control and case subjects a standard protocol has been used. SNP genotyping was performed using the MassARRAY SNP genotyping system (Sequenom Inc., San Diego, CA, USA) [47]. Re-genotyping of random samples, which represented a total of 600 successfully replicated genotypes, revealed an accuracy rate of >99.9%.

3.2. Datasets

Seven datasets have been used containing the following 48 SNPs at the DRD3 and HTR2A genes associated with schizophrenia from the Galician population [9]: rs4682148, rs7631540, rs6808291, rs1486012, rs9824856, rs2134655, rs963468, rs3773678, rs167771, rs226082, rs10934256, rs1486009, rs6280, rs7638876, rs9825563, rs1354348, rs9283560 (17 SNPs at DRD3) and rs3889066, rs7329640, rs10507544, rs7333412, rs3125, rs6314, rs6308, rs1058576, rs6561333, rs1923884, rs2296972, rs9316233, rs659734, rs1928042, rs2770296, rs9316235, rs582385, rs1928040, rs731779, rs985934, rs9534505, rs6304, rs6305, rs2070036, rs2070037, rs6313, rs1328685, rs731244, rs1360020, rs10507546, rs10507547 (31 SNPs at HTR2A). SNPs can take different values: 0 if homozygous (both copies of a given gene have the same allele) for the first allele (one of a number of alternative forms of the same gene occupying a given position on a chromosome), 1 if heterozygous (the patient has two different alleles of a given gene), 2 if homozygous for the second allele or unknown.

Additional negative subjects have been generated using the simulation tool named HAP-SAMPLE [15]. HAP-SAMPLE is a web application for simulating SNP genotypes for case-control and affected-child trio studies by re-sampling from Phase I/II HapMap SNP data. Providing a list of SNPs to be "genotyped," along with a disease model file that describes causal SNPs and their effect sizes, the

application returns two sets of simulated genotypes from case and control subjects. We discarded the case subjects. Thus, a file was created with a different number of control subjects, which were added to case subjects from real clinical data. This data was modified in order to introduce genotyping errors taking into account the error frequencies of the real data.

In addition to the original dataset that contains 260 positive subjects and 354 negative subjects SNP (1:0), we obtained six datasets by including 307, 614, 1,228, 1,842, 2,456 and 3,070 simulated negative subjects. The datasets were named: SNP (1:0.5), SNP (1:1), SNP (1:2), SNP (1:3), SNP (1:4) and SNP (1:5), where the label represents the proportion between the real subjects (positive and negative) and the simulated negative subjects.

3.3. QGDR models

The classification models have been obtained with the following methods: Linear Neural Networks, Multilayer Perceptron, Radial Base Functions, Bayesian Networks, Naïve Bayes, Support Machine Vectors, Decision Tables, Decision Table Naïve Bayes Hybrid Classifier, Best-First decision Tree classifier, Adaptative Boosting (all of them from Weka 3.6.2 [48]), Evolutionary Computation and Multifactor Dimensionality Reduction.

Artificial Neural Networks (ANN) have been extensively used for classification problems. More specifically, the simple Perceptron [49], also known as Linear Neural Network (LNN), has been utilized. This technique uses a linear network model, with no hidden layers, to perform classification. The Multilayer Perceptron (MLP) [50] has also been utilized. Other types of networks considered were Radial Base Functions (RBF) [51]. In this type of network, the neurons of the hidden layer perform a calculation function instead of the activation function of the MLP. The general scheme for an ANN with only one hidden layer is presented in Figure 5.

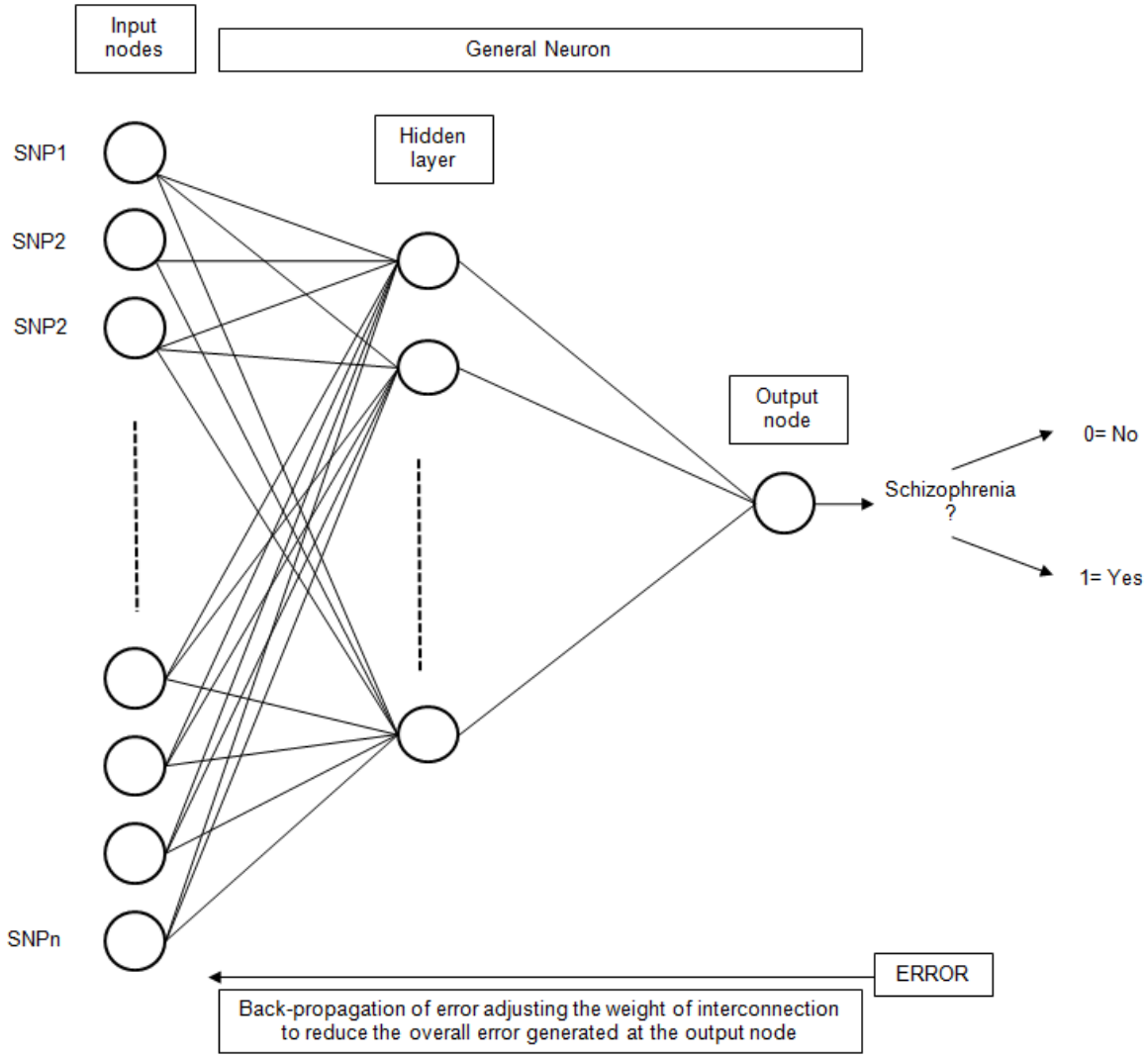
As well as the MLP, Support Machine Vectors (SVM) are nonlinear classifiers. SVM induce linear separators or hyperplanes in the space of characteristics. This type of classifier has proved to be very useful when dealing with high dimensionality problems [45].

Bayesian methods have also been applied to this problem. These methods are based on Bayes' theory of probability. Not only they allow performing classification, but they also allow finding relationships among attributes. Several of these methods have been used, such as Naive Bayes [52] (which assumes that the attributes are independent), and Bayesian Networks [53].

The following techniques allow obtaining classification models based on "IF-THEN-ELSE" rules or on hierarchical structures such as trees. More specifically, rule inference models from Decision Tables [54] have been used, building a decision table majority classifier. This type of method evaluates feature subsets using best-first search and uses the nearest-neighbor method to determine the class for each instance that is not covered by the decision table or by the Decision Table Naïve Bayes Hybrid Classifier (DTNB) DTNB [55]. A similar model was used to infer decision trees, following a hybrid approach between the decision trees and the Naïve Bayes classifier, called Best-First decision Tree classifier (BFTree) [56].

Finally, we tried a boosting meta-algorithm. This algorithm consists in combining multiple classification models that complement each other. The Adaptative Boosting (AdaBoost) [57] method builds the models iteratively, weighting the instances differently in each iteration. The new models classify the instances that the previous models do not classify correctly.

Figure 5. The general structure of an ANN for schizophrenia classification based on SNP inputs.



Multifactor Dimensionality Reduction (MDR) [58,59] is a data mining approach designed to detect and characterize nonlinear interactions among discrete attributes or variables that influence a binary outcome (for example, case-control status). It is a constructive induction algorithm which reduces the original n-dimensional model to a one-dimensional model, repeating this procedure for each possible n-factor combination and selecting the combination that maximizes the case-control ratio of the high-risk group. This method is considered to be a nonparametric alternative to traditional statistical methods. The MDR software combines attribute selection, attribute construction and classification with cross-validation. This method has mostly been used to detect gene-gene interactions or epistasis in genetic studies of common human diseases [60-62] such as schizophrenia [63-65], although it can also be applied to other domains.

The technique of Evolutionary Computation (EC) [46] used in this paper is based on genetic algorithms (GAs) [66]. A GA is a search method based on Charles Darwin's Theory of Evolution [67]. Algorithms based on GAs make a population evolve through random actions similar to those existing in biological evolution (mutations and genetic recombination, as well as selections with a certain

criteria called fitness). The fitness is used to decide which individuals are selected, *i.e.*, the more suitable individuals are the higher likelihood they will reproduce. More specifically, the method considered here follows the Iterative Rule Learning (IRL) approach [68,69]. Thus, the result of this method is a set of rules which are used to classify the input data. Like MDR, this method tries to find relationships between attributes or variables and a binary outcome. It has mostly been applied to biomedical data; however, it is still in development.

For each classification, the data has been split into two groups: *Schizo* (positive/case subjects) and *non-Schizo* (negative/control subjects). The SNPs have categorical values of "0" if homozygous for the first allele, "1" if heterozygous, "2" if homozygous for the second allele "3" for unknown genotypes. The 10-fold cross-validation method [70-72] has been used to verify the accuracy of the models. The efficiency of the models that evaluate if a patient has schizophrenia is mainly due by the number of correct classifications when using the test set. In addition, these models have been constructed using the SNPs at only one of the two genes or at both of them. Therefore, the classification results have been obtained using 12 machine learning techniques and seven datasets that include different percentages of simulated negative subjects, that is, 252 classification models to be tested.

4. Conclusions

This work presents a disease computational study of schizophrenia based on DNA molecule information provided by SNPs and proposes for the first time, to the best knowledge of the authors, two classification models for schizophrenia evaluation. 252 classification models have been obtained using SNPs at two schizophrenia-related genes (each of them or both), twelve machine learning techniques and seven datasets. The best relationships between the DNA molecule sequence and schizophrenia evaluated 78.3–93.8% of the DNA sequence from schizophrenia patients, for datasets with extra simulated negative subjects. In future work, QGDR models will be extended to other types of complex diseases, such as colorectal cancer and cardiovascular diseases, and the best models will be implemented online for free access.

Acknowledgements

The work of Vanessa Aguiar-Pulido is supported by a grant from the General Directorate of Quality and Management of Galicia's University System of the Xunta. Munteanu C. R. and José A. Seoane acknowledge the funding support for a research position by "Isidro Parga Pondal" Program and an "Isabel Barreto" grant from Xunta de Galicia (Spain), respectively. This work is supported by the "Galician Network for Colorectal Cancer Research" (REGICC, Ref. 2009/58), from the General Directorate of Scientific and Technologic Promotion of the Galician University System of Xunta de Galicia, by the "Ibero-American Network of the Nano-Bio-Info-Cogno Convergent Technologies", Ibero-NBIC Network (209RT0366) funded by CYTED (Ciencia y Tecnología para el Desarrollo) and by the COMBIOMED Network, the grant (Ref. PIO52048 and RD07/0067/0005), funded by the Carlos III Health Institute.

References

1. Chinchilla Moreno, A. *Las Esquizofrenias. Sus Hechos Y Valores Clínicos Y Terapéuticos*; Elsevier Masson: Barcelona, Spain, 2007.
2. Sham, P. Genetic epidemiology. *Br. Med. Bull.* **1996**, *52*, 408-433.
3. Sáiz, J.; Fañanás, L. Introducción: Genética y Psiquiatría. *Monogr. Psiquiatr.* **1998**, *10*, 1-3.
4. Meltzer, H.Y.; Matsubara, S.; Lee, J.C. Classification of typical and atypical antipsychotic drugs on the basis of dopamine D-1, D-2 and serotonin2 pKi values. *J. Pharmacol. Exp. Ther.* **1989**, *251*, 238-246.
5. Sokoloff, P.; Levesque, D.; Martres, M.P.; Lannfelt, L.; Diaz, G.; Pilon, C.; Schwartz, J.C. The dopamine D3 receptor as a key target for antipsychotics. *Clin. Neuropharmacol.* **1992**, *15*, 456A-457A.
6. Utsunomiya, K.; Shinkai, T.; De Luca, V.; Hwang, R.; Sakata, S.; Fukunaka, Y.; Chen, H.I.; Ohmori, O.; Nakamura, J. Genetic association between the dopamine D3 gene polymorphism (Ser9Gly) and schizophrenia in Japanese populations: evidence from a case-control study and meta-analysis. *Neurosci. Lett.* **2008**, *444*, 161-165.
7. Suzuki, M.; Hurd, Y.L.; Sokoloff, P.; Schwartz, J.C.; Sedvall, G. D3 dopamine receptor mRNA is widely expressed in the human brain. *Brain Res.* **1998**, *779*, 58-74.
8. Talkowski, M.E.; Mansour, H.; Chowdari, K.V.; Wood, J.; Butler, A.; Varma, P.G.; Prasad, S.; Semwal, P.; Bhatia, T.; Deshpande, S.; Devlin, B.; Thelma, B.K.; Nimgaonkar, V.L. Novel, replicated associations between dopamine D3 receptor gene polymorphisms and schizophrenia in two independent samples. *Biol. Psychiat.* **2006**, *60*, 570-577.
9. Dominguez, E.; Loza, M.I.; Padin, F.; Gesteira, A.; Paz, E.; Paramo, M.; Brenlla, J.; Pumar, E.; Iglesias, F.; Cibeira, A.; Castro, M.; Caruncho, H.; Carracedo, A.; Costas, J. Extensive linkage disequilibrium mapping at HTR2A and DRD3 for schizophrenia susceptibility genes in the Galician population. *Schizophr. Res.* **2007**, *90*, 123-129.
10. den Dunnen, J.T.; Antonarakis, S.E. Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion. *Hum. Mutat.* **2000**, *15*, 7-12.
11. Katanforoush, A.; Sadeghi, M.; Pezeshk, H.; Elahi, E. Global haplotype partitioning for maximal associated SNP pairs. *BMC Bioinformatics* **2009**, *10*, 269.
12. Zhou, N.; Wang, L. Effective selection of informative SNPs and classification on the HapMap genotype data. *BMC Bioinformatics* **2007**, *8*, 484.
13. Kingman, J. F. Origins of the coalescent. 1974-1982. *Genetics* **2000**, *156*, 1461-1463.
14. Liang, L.; Zollner, S.; Abecasis, G.R. GENOME: a rapid coalescent-based whole genome simulator. *Bioinformatics* **2007**, *23*, 1565-1567.
15. Wright, F.A.; Huang, H.; Guan, X.; Gamiel, K.; Jeffries, C.; Barry, W.T.; de Villena, F.P.; Sullivan, P.F.; Wilhelmsen, K.C.; Zou, F. Simulating association studies: a data-based resampling method for candidate regions or whole genome scans. *Bioinformatics* **2007**, *23*, 2581-2588.
16. Balloux, F. EASYPOP (version 1.7): a computer program for population genetics simulations. *J. Hered.* **2001**, *92*, 301-302.
17. Hey, J. *FPG: A Computer Program for Forward Population Genetic Simulation*, 2004. http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm#FPG/ (accessed on 5 May 2010).

18. Hoggart, C.J.; Chadeau-Hyam, M.; Clark, T.G.; Lampariello, R.; Whittaker, J.C.; De Iorio, M.; Balding, D.J. Sequence-level population simulations over large genomic regions. *Genetics* **2007**, *177*, 1725-1731.
19. Peng, B.; Kimmel, M. simuPOP: a forward-time population genetics simulation environment. *Bioinformatics* **2005**, *21*, 3686-3687.
20. Edwards, T.L.; Bush, W.S.; Turner, S.D.; Dudek, S.M.; Torstenson, E.S.; Schmidt, M.; Martin, E.; Ritchie, M.D. Generating Linkage Disequilibrium Patterns in Data Simulations using genomeSIMLA. *Lect. Notes Comput. Sci.* **2008**, *4973*, 24-35.
21. Li, J.; Chen, Y. Generating samples for association studies based on HapMap data. *BMC Bioinformatics* **2008**, *9*, 44.
22. Ban, H.J.; Heo, J.Y.; Oh, K.S.; Park, K.J. Identification of Type 2 Diabetes-associated combination of SNPs using Support Vector Machine. *BMC Genet.* **2010**, *11*, 26.
23. Saangyong, U; Dong-Hoi, K.; Young-Woong, K.; Sungwon, C; Jaeyoun, C.; Jin, K. A study on application of single nucleotide polymorphism and machine learning techniques to diagnosis of chronic hepatitis. *Expert Systems* **2009**, *26*, 60-69.
24. Briggs, F.B.; Ramsay, P.P.; Madden, E.; Norris, J.M.; Holers, V.M.; Mikuls, T.R.; Sokka, T.; Seldin, M.F.; Gregersen, P.K.; Criswell, L.A.; Barcellos, L.F. Supervised machine learning and logistic regression identifies novel epistatic risk factors with PTPN22 for rheumatoid arthritis. *Genes Immun.* **2010**, *11*, 199-208.
25. Nicodemus, K.K.; Callicott, J.H.; Higier, R.G.; Luna, A.; Nixon, D.C.; Lipska, B.K.; Vakkalanka, R.; Giegling, I.; Rujescu, D.; Clair, D.S.; Muglia, P.; Shugart, Y.Y.; Weinberger, D.R. Evidence of statistical epistasis between DISC1, CIT and NDEL1 impacting risk for schizophrenia: biological validation with functional neuroimaging. *Hum. Genet.* **2010**, *127*, 441-452.
26. Devillers, J.; Balaban, A.T. *Topological Indices and Related Descriptors in QSAR and QSPR*; Gordon and Breach: Amsterdam, The Netherlands, 1999.
27. Barabasi, A.L.; Bonabeau, E. Scale-free networks. *Sci. Am.* **2003**, *288*, 60-69.
28. Balaban, A. T.; Basak, S.C.; Beteringhe, A.; Mills, D.; Supuran, C.T. QSAR study using topological indices for inhibition of carbonic anhydrase II by sulfanilamides and Schiff bases. *Mol. Divers.* **2004**, *8*, 401-412.
29. Barabasi, A.L.; Oltvai, Z.N. Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **2004**, *5*, 101-113.
30. Barabasi, A.L. Sociology. Network theory-the emergence of the creative enterprise. *Science* **2005**, *308*, 639-641.
31. González-Díaz, H.; Vilar, S.; Santana, L.; Uriarte, E. Medicinal Chemistry and Bioinformatics – Current Trends in Drugs Discovery with Networks Topological Indices. *Curr. Top. Med. Chem.* **2007**, *7*, 1025-1039.
32. Ferino, G.; Gonzalez-Diaz, H.; Delogu, G.; Podda, G.; Uriarte, E. Using spectral moments of spiral networks based on PSA/mass spectra outcomes to derive quantitative proteome-disease relationships (QPDRs) and predicting prostate cancer. *Biochem. Biophys. Res. Commun.* **2008**, *372*, 320-325.
33. Gonzalez-Diaz, H.; Gonzalez-Diaz, Y.; Santana, L.; Ubeira, F.M.; Uriarte, E. Proteomics, networks and connectivity indices. *Proteomics* **2008**, *8*, 750-778.

34. Munteanu, C.R.; Magalhaes, A.L.; Uriarte, E.; Gonzalez-Diaz, H. Multi-target QPDR classification model for human breast and colon cancer-related proteins using star graph topological indices. *J. Theor. Biol.* **2009**, *257*, 303-311.
35. Vilar, S.; Gonzalez-Diaz, H.; Santana, L.; Uriarte, E. A network-QSAR model for prediction of genetic-component biomarkers in human colorectal cancer. *J. Theor. Biol.* **2009**, *261*, 449-458.
36. Vilar, S.; Gonzalez-Diaz, H.; Santana, L.; Uriarte, E. QSAR model for alignment-free prediction of human breast cancer biomarkers based on electrostatic potentials of protein pseudofolding HP-lattice networks. *J. Comput. Chem.* **2008**, *29*, 2613-2622.
37. González-Díaz, H.; Ferino, G.; Prado-Prado, F.J.; Vilar, S.; Uriarte Villares, E.; Pazos, A.; Munteanu, C.R. Protein Graphs in Cancer Prediction. In *An Omics Perspective on Cancer Research*; Cho, W.C.S., Ed.; Springer Netherlands: Amsterdam, The Netherlands, 2010; doi:10.1007/978-90-481-2675-0_7.
38. González-Díaz, H.; Ferino, G.; Podda, G.; Uriarte, E. Discriminating Prostate Cancer Patients from control group with connectivity indices. *ECSOC* **2008**, *12*, G1:1-G1:10.
39. Ferino, G.; Delogu, G.; Podda, G.; Uriarte, E.; González-Díaz, H. Quantitative Proteome-Disease Relationships (QPDRs) in Clinical Chemistry: Prediction of Prostate Cancer with Spectral Moments of PSA/MS Star Networks. In *Clinical Chemistry Research*; Mitchem, B.H., Sharnham, C.L. Eds.; Nova Science Publishers: New York, NY, USA, 2009.
40. Diederich, J. *Artificial Neural Networks: Concept Learning*; IEEE Press: Piscataway, NJ, USA, 1990; p. 141.
41. Byvatov, E.; Schneider, G. Support vector machine applications in bioinformatics. *Appl. Bioinformatics* **2003**, *2*, 67-77.
42. Eberbach, E. Toward a theory of evolutionary computation. *Biosystems* **2005**, *82*, 1-19.
43. Rowland, J.J. Model selection methodology in supervised learning with evolutionary computation. *Biosystems* **2003**, *72*, 187-196.
44. Tan, P.-N.; Steinbach, M.; Kumar, V. *Introduction to Data Mining*; Pearson Addition Wesley: Boston, ML, USA, 2006.
45. Vapnik, V. *Statistical Learning Theory*; John Weily and Sons: New York, NY, USA, 1998.
46. Aguiar Pulido, V.; Seoane Fernández, J.A.; Freire, A.; Munteanu, C.R. Data Mining in Complex Diseases Using Evolutionary Computation. *Lect. Notes Comput. Sci.* **2009**, *5517*, 917-924.
47. Costas, J.; Torres, M.; Cristobo, I.; Phillips, C.; Carracedo, A. Relative efficiency of the linkage disequilibrium mapping approach in detecting candidate genes for schizophrenia in different European populations. *Genomics* **2005**, *86*, 280-286.
48. Waikato, T.U.O. Weka Machine Learning Project. http://www.cs.waikato.ac.nz/ml/weka/ (accessed on 5 May 2010).
49. Rosenblatt, F. *Principles of Neurodynamics; Perceptrons and The Theory of Brain Mechanisms*; Spartan Books: Washington, DC, USA, 1962.
50. Bishop, C. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, NY, USA, 1995.
51. Buhmann, M.D. *Radial Basis Functions: Theory and Implementations*; Cambridge University Press: Cambridge, UK, 2003.

52. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of 11th Conference on Uncertainty in Artificial Intelligence*, Montreal, Quebec, August 18-20, 1995; Morgan Kaufmann: San Mateo, CA, USA, 1995; pp. 338-345.
53. Bouckaert, R.R. *Bayesian Networks in Weka*; Technical report, Computer Science Department. University of Waikato: Tauranga, New Zealand, 2004.
54. Kohavi, R. The Power of Decision Tables. In *Proceedings of 8th European Conference on Machine Learning*, Heraclion, Greece, April 25-27, 1995; Levrac, N., Wrobel, S., Eds.; Springer-Verlag Publisher: London, UK, 1995; pp.174-189.
55. Mark Hall, E.F. Combining Naive Bayes and Decision Tables. In *Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS)*, Coconut Grove, Florida, May 15–17, 2008; AAAI Press: Menlo Park, CA, USA, 2008.
56. Shi, H. Best-first Decision Tree Learning. MSc Thesis, University of Waikato, Hamilton, New Zealand, 2007.
57. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, Desenzano sul Garda, Italy, June 28 to July 1, 1996; Saitta, L., Ed., Morgan Kaufmann: San Francisco, CA, 1996; pp. 148-156,
58. Moore, J.H.; Gilbert, J.C.; Tsai, C.T.; Chiang, F.T.; Holden, T.; Barney, N.; White, B.C. A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. *J. Theor. Biol.* **2006**, *241*, 252-261.
59. Cordell, H.J. Detecting gene-gene interactions that underlie human diseases. *Nat. Rev. Genet.* **2009**, *10*, 392-404.
60. Greene, C.S.; Sinnott-Armstrong, N.A.; Himmelstein, D.S.; Park, P.J.; Moore, J.H.; Harris, B.T. Multifactor dimensionality reduction for graphics processing units enables genome-wide testing of epistasis in sporadic ALS. *Bioinformatics* **2010**, *26*, 694-695.
61. Cattaert, T.; Urrea, V.; Naj, A.C.; De Lobel, L.; De Wit, V.; Fu, M.; Mahachie John, J.M.; Shen, H.; Calle, M.L.; Ritchie, M.D.; Edwards, T.L.; Van Steen, K. FAM-MDR: a flexible family-based multifactor dimensionality reduction technique to detect epistasis using related individuals. *PLoS One* **2010**, *5*, e10304.
62. He, H.; Oetting, W.S.; Brott, M.J.; Basu, S. Pair-wise multifactor dimensionality reduction method to detect gene-gene interactions in a case-control study. *Hum. Hered.* **2010**, *69*, 60-70.
63. Kang, S.G.; Lee, H.J.; Choi, J.E.; Park, Y.M.; Park, J.H.; Han, C.; Kim, Y.K.; Kim, S.H.; Lee, M.S.; Joe, S.H.; Jung, I.K.; Kim, L. Association Study between Antipsychotics - Induced Restless Legs Syndrome and Polymorphisms of Dopamine D1, D2, D3, and D4 Receptor Genes in Schizophrenia. *Neuropsychobiology* **2008**, *57*, 49-54.
64. Vilella, E.; Costas, J.; Sanjuan, J.; Guitart, M.; De Diego, Y.; Carracedo, A.; Martorell, L.; Valero, J.; Labad, A.; De Frutos, R.; Najera, C.; Molto, M.D.; Toirac, I.; Guillamat, R.; Brunet, A.; Valles, V.; Perez, L.; Leon, M.; de Fonseca, F. R.; Phillips, C.; Torres, M. Association of schizophrenia with DTNBP1 but not with DAO, DAOA, NRG1 and RGS4 nor their genetic interaction. *J. Psychiatr. Res.* **2008**, *42*, 278-288.

65. Yasuno, K.; Ando, S.; Misumi, S.; Makino, S.; Kulski, J.K.; Muratake, T.; Kaneko, N.; Amagane, H.; Someya, T.; Inoko, H.; Suga, H.; Kanemoto, K.; Tamiya, G. Synergistic association of mitochondrial uncoupling protein (UCP) genes with schizophrenia. *Am. J. Med. Genet. B. Neuropsychiatr. Genet.* **2007**, *144B*, 250-253.
66. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
67. Darwin, C. *On the Origin of Species by Means of Natural Selection*; John Murray: London, UK, 1859.
68. Venturini, G. SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts. In *Proceedings of the 6th European Conference on Machine Learning*, Vienna, Austria, April 5-7, 1993; Brazdil, P., Ed.; Springer Verlag: Vienna, Austria, 1993; pp. 280-296.
69. González, A.; Herrera, F. Multi-stage genetic fuzzy systems based on the iterative rule learning approach. *Mathware Soft Comput.* **1997**, *4*, 233-249.
70. McLachlan, G.J.; Do, K.-A.; Ambroise, C. *Analyzing Microarray Gene Expression Data*. Wiley-Interscience: Hoboken, NJ, USA, 2004.
71. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 20-25, 1995; Morgan Kaufmann Publisher: San Francisco, CA, USA, 1995; Volume 2, pp. 1137-1143.
72. Picard, R.; Cook, D. Cross-Validation of Regression Models. *J. Amer. Statist. Assn.* **1984**, *79*, 575–583.

Send Orders of Reprints at reprints@benthamscience.net

Current Topics in Medicinal Chemistry, 2013, *13*, 000-000

Applied Computational Techniques on Schizophrenia Using Genetic Mutations

Vanessa Aguiar-Pulido, Marcos Gestal*, Carlos Fernandez-Lozano, Daniel Rivero and Cristian R. Munteanu

Information and Communications Technologies Department, Computer Science Faculty, University of A Coruña, Campus de Elviña s/n, 15071 Spain

Abstract: Schizophrenia is a complex disease, with both genetic and environmental influence. Machine learning techniques can be used to associate different genetic variations at different genes with a (schizophrenic or non-schizophrenic) phenotype. Several machine learning techniques were applied to schizophrenia data to obtain the results presented in this study. Considering these data, Quantitative Genotype – Disease Relationships (QDGRs) can be used for disease prediction. One of the best machine learning-based models obtained after this exhaustive comparative study was implemented online; this model is an artificial neural network (ANN). Thus, the tool offers the possibility to introduce Single Nucleotide Polymorphism (SNP) sequences in order to classify a patient with schizophrenia. Besides this comparative study, a method for variable selection, based on ANNs and evolutionary computation (EC), is also presented. This method uses half the number of variables as the original ANN and the variables obtained are among those found in other publications. In the future, QDGR models based on nucleic acid information could be expanded to other diseases.

Keywords: Bioinformatics, data mining, machine learning, neural networks, schizophrenia, SNP, support vector machines.

1. INTRODUCTION

The study of diseases with computational models uses different molecular information such as structure and physical/chemical properties of a protein and DNA/RNA molecules, blood proteome mass spectra, DNA microarrays, disease biomarkers and concentration of the metabolites in physiological liquids. Schizophrenia is a common mental disorder defined as a heterogeneous syndrome characterized by perturbations in language, perception, thinking, social relationships and will as a consequence of several cumulative effects of certain (genetic and environmental) risk factors [1] or epigenetics [2]. Due to the impact of this disease, molecular genetics techniques have been used to identify the genes related with this disorder.

The computational methods are focused on finding the relationships between schizophrenia and molecular information. Quantitative Structure – Activity Relationships (QSARs) are widely used for predicting protein properties [3] and Quantitative Protein (or Proteome) – Disease Relationships (QPDRs) [4-10] for disease prediction. The numerical data used for these classifications consisted in topological indices or molecular descriptors obtained with the Graph/Complex Network theory [11-14]. Several QSAR/QPDR models based on protein structure and proteome mass spectra have been obtained for cancer [15-18], especially for breast and colorectal cancer [19, 20] and prostate cancer [21]. Additional applications have been published for protein interactions in parasites [22-24].

*Address correspondence to this author at the Information and Communications Technologies Department, Computer Science Faculty, University of A Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain; Tel: (+34) 98167000; Ext: 1379; Fax: (+34) 981167160; E-mail: mgestal@udc.es.

In a similar way, a QGDR can be established in order to automatically evaluate schizophrenia DNA sequences using Single Nucleotide Polymorphisms (SNP) data. A SNP [25] is a single nucleotide variation in a genetic sequence that occurs at appreciable frequency in the population, that is, at least in 1%. Thus, SNPs can be used as inputs in disease computational studies such as pattern searching or classification models. Models based on machine learning have been extensively used to analyse complex diseases, such as diabetes [26], hepatitis [27] and rheumatoid arthritis [28]. However, not many studies have been carried out on variation analysis in schizophrenia using Machine Learning algorithms [29, 30]. Statistical models were the most used for this type of complex disease.

Ban *et al.* [26] analyzed the importance of gene-gene interactions in Type 2 diabetes mellitus (T2D) susceptibility by investigating 408 SNPs in 87 genes involved in major T2D-related pathways in 462 T2D patients and 456 healthy controls from the Korean cohort studies. They used the support vector machine (SVM) method to differentiate between cases and controls using SNP information in a 10-fold cross-validation test and they achieved a 65.3% prediction rate with a combination of 14 SNPs in 12 genes by using the radial basis function (RBF)-kernel SVM. As the high-throughput technology for genome-wide SNPs improves, it is likely that a much higher prediction rate with biologically more interesting combination of SNPs can be acquired by using this method. Thus, SVM-based feature selection method in this study found novel association between combinations of SNPs and T2D in a Korean population.

Uhmn *et al.* [27] used several machine learning techniques to predict the susceptibility to chronic hepatitis from

1568-0266/13 $58.00+.00

SNP data, integrated with several feature selection algorithms to identify a set of SNPs relevant to the disease. They applied a backtracking technique to a couple of feature selection algorithms, forward selection and backward elimination, and showed that it was beneficial to find the best solutions by experiment. The experimental results show that the decision rule was able to distinguish between chronic and normal hepatitis with a maximum accuracy of 73.20%, whereas the accuracy of the support vector machine was 67.53% and that of the decision tree was 72.68%. It was also shown that the decision tree and decision rule are potential tools to predict susceptibility to chronic hepatitis from SNP data.

Briggs *et al.* [28] studied the genetic interactions (epistasis) with a statistical approach, by combining several analytical methods. Thus, they used a multi-stage analysis that incorporated supervised machine learning and methods of association testing, to investigate epistatic interactions with a well-established genetic factor (PTPN22 1858T) in a complex autoimmune disease such as rheumatoid arthritis (RA). The analysis consisted of four principal stages: Stage I (data reduction) - identifying candidate chromosomal regions in 292 affected sibling pairs, by predicting PTPN22 concordance using multipoint identity-by-descent probabilities and a supervised machine learning algorithm (Random Forests); Stage II (extension analysis) - testing detailed genetic data within candidate chromosomal regions for epistasis with PTPN22 1858T in 677 cases and 750 controls using logistic regression; Stage III (replication analysis) - confirmation of epistatic interactions in 947 cases and 1,756 controls; Stage IV (combined analysis) - a pooled analysis including all 1624 RA cases and 2,506 control subjects for final estimates of effect size. A total of seven replicating epistatic interactions were identified. The results demonstrate that the SNP variants within CDH13, MYO3A, CEP72 and near WFDC1 showed significant evidence for interaction with PTPN22, affecting susceptibility to RA.

One of the most studied genes related to schizophrenia susceptibility is DRD3. Same as HTR2A, it is considered to be an important target for several antipsychotic drugs [31]. HTR2A encodes one of the receptors for serotonin and DRD3 encodes one subtype of the five dopamine receptors, both neurotransmitters. More specifically, Dopamine 3 receptors (DRD3) are concentrated in limbic regions of the brain, which are associated with cognitive, emotional and endocrine functions. Thus, it may be particularly relevant to schizophrenia, as the DRD3 messenger RNA is predominantly expressed in the limbic system, a region thought to be dysfunctional in this disease [32]. Association studies involving these functional candidate genes have systematically focused on a limited set of SNPs, generally based on previously reported small contributions of these markers of risk of susceptibility to schizophrenia. More specifically, SNP T102C (rs6313) at HTR2A and SNP Ser9Gly (rs6280) at DRD3 have been extensively analyzed in several schizophrenia case-control studies [33].

The serotonin transporter gene (SLC6A4) and its promoter (5-HTTLPR) polymorphism have been the focus of a large number of association studies of behavioral traits and psychiatric disorders such as schizophrenia. However, large-scale genotyping of the polymorphism has been very difficult. Lu *et al.* [30] reported the development and validation of a 5-HTTLPR genotype prediction model. The single nucleotide polymorphisms (SNPs) from the 2,000 kb region surrounding 5-HTTLPR were used to construct a prediction model through a newly developed machine learning method, multicategory vertex discriminant analysis with 2,147 individuals from the Northern Finnish Birth Cohort genotyped with the Illumina 370K SNP array and manually genotyped for 5-HTTLPR polymorphism. The prediction model was applied to SNP genotypes in a Dutch/German schizophrenia case-control sample of 3,318 individuals to test the association of the polymorphism with schizophrenia. The prediction model of eight SNPs achieved a 92.4% accuracy rate and a 0.98±0.01 area under the receiving operating characteristic. Thus, evidence for an association of these SNPs with schizophrenia was observed (P=0.05, odds ratio=1.105). This prediction model provides an effective substitute of manually genotyped 5-HTTLPR alleles, providing a new approach for large scale association studies of this polymorphism.

The current review will present details about the comparative study of machine learning disease classification models using only SNPs at the HTR2A and DRD3 genes in Galician (Northwestern Spain) schizophrenic patients. Methods such as ANNs [34], SVMs [35-37], EC [38-40] and other machine learning techniques [41] have been used to find the best classification models.

Once this comparison was finished, the machine learning-based method which obtains the best results in Ref. [42] was implemented online as SNP-Schizo (http://bio-aims.udc.es/SNPSchizo.php) in the Bio-AIMS server. This tool also includes an approach for variable selection, based on ANNs and evolutionary computation (EC).

2. MATERIALS AND METHODS

Fig. **1** summarizes the workflow followed by this approach. Firstly, data from patients is genotyped in order to obtain SNP sequences. After that, computational methods are applied to this data in order to obtain QGDR classification models. Finally, the models obtained are evaluated using new data. Thus, this procedure allows establishing relationships between SNP sequences and the predisposition to the disease.

2.1. Schizophrenia Data

For the comparative study, schizophrenia data collected from Galician patients [33] were used. These data contained 48 SNPs at the DRD3 and HTR2A genes, which are associated to schizophrenia. These SNPs were encoded taking different values:

- 0 if homozygous (both copies of a given gene have the same allele) for the first allele (one of a number of alternative forms of the same gene occupying a given position on a chromosome),
- 1 if heterozygous (the patient has two different alleles of a given gene),
- 2 if homozygous for the second allele or
- 3 if unknown.

The original dataset contained 260 positive subjects (genetically predisposed to schizophrenia) and 354 negative subjects (not predisposed), a total of 614 patients.

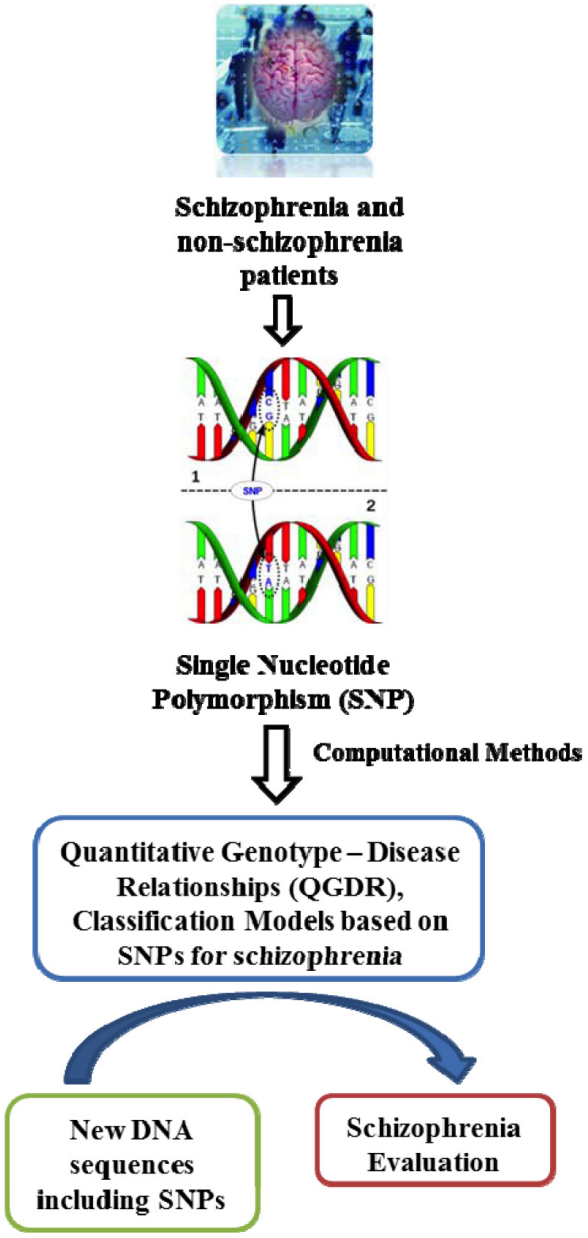


Fig. (1). QGDR Model classification.

To perform further tests, six other datasets were obtained from the original one. This was carried out by adding negative subjects generated with the HAP-SAMPLE [43] simulation tool. These data were modified to include genotyping errors (represented as value 3) taking into account the error frequencies of the real data, but choosing randomly which positions were modified. Thus, these datasets included 307, 614, 1,228, 1,842, 2,456 and 3,070 simulated negative subjects. Datasets were named following the pattern 1:N, where this label represents the proportion between the real subjects (positive and negative) and the simulated negative subjects.

However, there are several genetic data simulation packages. Among those, we consider the coalescent-based methods [44], which have been used for population-based simulation in genetic studies, such as GENOME [45]. This method was developed to overcome previous limitations. HAP-SAMPLE [43], which is the simulator used in this paper, uses the existing Phase I/II HapMap data to resample existing phased chromosomes to simulate datasets. There are also forward-time population simulations, such as easyPOP [46], FPG [47], FREGENE [48], simuPOP [49] and genomeSIMLA [50]. The last method can simulate realistic patterns of LD in both family-based and case-control datasets and, unlike other similar packages, has proved to be an effective platform for simulating large scale genetic data. Another program capable of generating large scale genetic as well as phenotypic variation data is presented in ref. [51]. This program generates genotypes/phenotypes by perturbing real data, with the aim of creating a large number of replicates that share similar properties with real data. Nevertheless, since HAP-SAMPLE is an association simulator for candidate regions and was specifically designed for simulating SNP genotypes for case-control studies, it was the most appropriate simulator given the nature of the original data used in this study.

2.2. Machine Learning Methods

ANNs [52, 53] have been extensively used for classification problems. More specifically, a multilayer Linear Neural Network (LNN) has been used. This technique uses a linear network model, as the activation functions is linear, and always has an equivalent single layer counterpart [54]. The Multilayer Perceptron (MLP) [52] has also been used. Other types of networks considered were the Radial Base Functions (RBF). In this type of network, the neurons of the hidden layer perform a calculation function instead of an activation function of the MLP.

Same as the MLP, SVMs belongs to non-linear classifiers. SVMs induce linear separators or hyperplanes in the space of characteristics. This type of classifier has proved to be very useful when dealing with high dimensionality problems [55].

Bayesian methods are based on Bayes' theory of probability. Not only they allow performing classification, but they also allow finding relationships among attributes. Several of these methods have been used, such as Naive Bayes [56] (which assumes that the attributes are independent) and Bayesian Networks [57].

The following techniques allow obtaining classification models based on "IF-THEN-ELSE" rules or on hierarchical structures such as trees. More specifically, rule inference models from Decision Tables [58] are obtained by building a decision table majority classifier. This type of method evaluates feature subsets using best-first search and uses the nearest-neighbor method to determine the class for each instance that is not covered by the decision table or by the Decision Table Naive Bayes Hybrid Classifier (DTNB) [59]. A similar model was considered to infer decision trees, following a hybrid approach between the decision trees and the Naive Bayes classifier, called Best-First decision Tree classifier (BFTree) [60].

A boosting meta-algorithm was also included in this study. This algorithm consists of combining multiple classification models that complement each other. The Adaptive Boosting (AdaBoost) [61] method builds the models iteratively, weighing the instances differently in each iteration. The new models classify the instances that the previous models did not classify correctly.

Multifactor Dimensionality Reduction (MDR) [62, 63] is a data mining approach designed to detect and characterize non-linear interactions among discrete attributes or variables that influence a binary outcome (for example, case-control status). It is a constructive induction algorithm which reduces the original n-dimensional model to a one-dimensional model, repeating this procedure for each possible n-factor combination and selecting the combination that maximizes the case-control ratio of the high-risk group. This method is considered to be a non-parametric alternative to traditional statistical methods. The MDR software combines attribute selection, attribute construction and classification with cross-validation. This method has mostly been used to detect gene-gene interactions or epistasis in genetic studies of common human diseases [64-66] such as schizophrenia [67-69], although it can also be applied to other domains.

2.3. Improving Machine Learning Methods by Means of Variable Selection

Once the comparative study is presented, a novel approach based on a previous variable selection will be discussed. This new approach uses Genetics Algorithms (GA) and ANNs in a first stage to establish which are the most relevant variables within the data. In the second stage, the classification stage, ANNs and SVMs will be used.

ANN and GA for Variable Selection

GAs [70-72] represent a search method based on Charles Darwin's Theory of Evolution [73]. This algorithm makes a population evolve through random actions similar to those existing in biological evolution such as mutations and genetic recombination, as well as selections with a certain criterion called fitness. The fitness is used to decide which individuals are selected, i.e., the most suitable individuals are those with the higher likelihood they will reproduce. Thus, the result of this method is a set of rules which are used to classify the input data. Thus, this method tries to find relationships between attributes or variables and a binary outcome [74-77].

ANN-GA approach [38, 39] uses "pruned" search, which starts by considering all the variables and gradually discards groups of them. The remaining set of variables is used to classify the samples, and the results are used to determine how relevant the discarded variables were for the classification. This process can be continued as long as the classification results are equal, or at least similar, to those obtained using the overall set of variables. Therefore, the GA determines how many and which variables will be considered for the classification. An ANN was included within the GA to evaluate the fitness values of the individuals. The use of an 'inner' ANN to evaluate fitness avoids definition and optimization of more formal equations and, remarkably, yields generality to the approaches presented herein. As the goal is to determine which solutions, out of those provided by the GA, represent good starting points to get acceptable classification models, it is not required to fully train such ANN; instead, extending the training up to the point where the ANN starts converging is enough.

In other words, the pruned search consists of a stepwise approach by which the GA steadily reduces the number of variables characterizing the samples, until an optimal subset is obtained. Each individual in the genetic population is initially described by d genes, each representing an original variable (using a binary encoding, each gene can be either 0 or 1).

Fitness will guide the pruning process (a black-box approach) to get individuals that, besides classifying as accurately as possible, use less variables. Eq. (1) defines how fitness can be described according to two parameters: the number of variables used to classify the samples and the quality of their classifications, calculated using the Mean Square Error (MSE) of the inner-ANN. Eq. (2) (employed here) shows that fitness will favor those individuals with less active genes (the denominator being the total number of variables).

$$\text{fitness}(\text{individual}_i) = f(\text{classification}_i) + f(\text{Selected variables}) \quad (\mathbf{1})$$

$$\text{fitness}(\text{individual}_i) = \text{MSE}(\text{ANN}_i) + \#1\text{'s genotype individual}_i / \#\text{total variables} \quad (\mathbf{2})$$

A good characteristic of Eqs. (1) and (2) is that they can be tailored. For instance, fitness may consider the cardinality (i.e., number of variables that have been selected for classification) or the percentage of variables (regarding the overall initial set of variables) being used.

Similar approaches were applied in the diagnosis of dermatological diseases [78], prediction of outcome [79] or heart problems [80] among others. In other fields this kind of approaches are also widely used [81-83]

ANN and SVM for Classification

After the variables have been selected, a classification algorithm has been applied in order to build the classification model. In the variables selection phase, a simple ANN model was built for the fitness score, but in the classification phase, this ANN has been replaced by a more complex model with more complex training. Several models have been tested, mainly SVMs and different ANN models. These ANN and SVM models have been developed using the Weka software [84], specifically, the MLP and Sequential Minimal Optimization (SMO) algorithm implementations.

3. RESULTS

3.1. Comparative Study

252 QDGR classification models were obtained after applying machine learning techniques to the data described previously. Seven datasets were used. The Weka software package [84] was used to perform the comparative study. This work presents the results achieved with the best responses provided by the most representative algorithms included in this software. In addition to LNN, the following techniques were applied to the datasets: MLP, RBF, EC,

MDR, Naive Bayes, Bayes Networks, SVM, Decision Tables, DTNB, BFTree and AdaBoost.

After carrying out this comparative study, the neural network model was implemented online. This approach consists of a type of ANN, hereafter referred to as LNN, that has a linear activation function in all neurons. More specifically, it is a multilayer neural network, with 40 neurons in the first layer, 152 in the second layer and 1 neuron as output. The number of input neurons was selected according to the results obtained from several feature selection methods (Best First [85], Linear Forward Selection [86], FCBF Search [87], Genetic Search [88], Scatter Search [89] and Random Search [90]). After several runs, it was proved that taking as input only the 40 neurons selected by the previous selection methods, the method achieved good results.

A graphical representation of the evolution of the different methods is shown in (Fig. **2**). As said before, 1:N represents the proportion between the real subjects and the simulated negative subjects. Thus, the first dataset does not include any simulated subject and the last dataset includes 5 simulated subjects per real one.

For each method, the percentage of correctly classified subjects is shown for each dataset. It can be observed that the classification percentages do not increase significantly after adding five parts of simulated subjects. Thus, we will focus on the results obtained for the dataset which contains the lowest number of simulated subjects, that is, the 1:0.5 dataset.

Classification accuracy percentages range from 56.6 to 66.6% for 1:0, which is the original dataset. For the datasets which included simulated subjects, these percentages range from 60.5 to 78.2% for 1:0.5, 69.8 to 83.0% for 1:1, 76.2 to 88.8% for 1:2, 84.8 to91.5% for 1:3, 87.4 to 93.2% for 1:4 and 88.4 to 94.3% for 1:5.

Among the best models, the LNN described above is proposed. This QGDR model includes only a minimum of simulated subjects (1:0.5). Thus, this dataset is made up of 921 subjects: 260 real positive subjects, 354 real negative subjects and 307 simulated negative subjects for schizophrenia. As mentioned previously, this neural network is based on 40 SNPs which are taken as an input and it has a hidden layer of 152 neurons. This technique obtained 78.2% in test accuracy when the 1:0.5 dataset was used as input.

The LNN achieves good results for all the datasets, as it is simpler and less computationally expensive than other methods. In (Fig. **3**), the area under the receiver operating curve (AUC-ROC) for the cross-validation group (0.8405) demonstrates that this model is not a random one. In addition, for this model, the threshold is 0.8.

3.2. Results After Variable Selection

It is important to point out that in order to characterize a complex disease, 40 SNPs are too many. Therefore, another approach based on genetic algorithms, artificial neural networks and SVMs, which was described previously, was applied to the 1:0.5 dataset. The best SVM classification model was a Weka SMO implementation with a complexity of 5, building logistics models [91], using a polynomial kernel [84]. Thus, the model is based on a LNN 40:152-1-1, data set 1:0.5 and it has a test accuracy of 78.2%.

This method obtains similar results to the LNN in terms of classification scores and AUC-ROC values (Table **1**) using less than half of the variables: only 17 variables were considered, instead of the 40 variables required by the previous LNN.

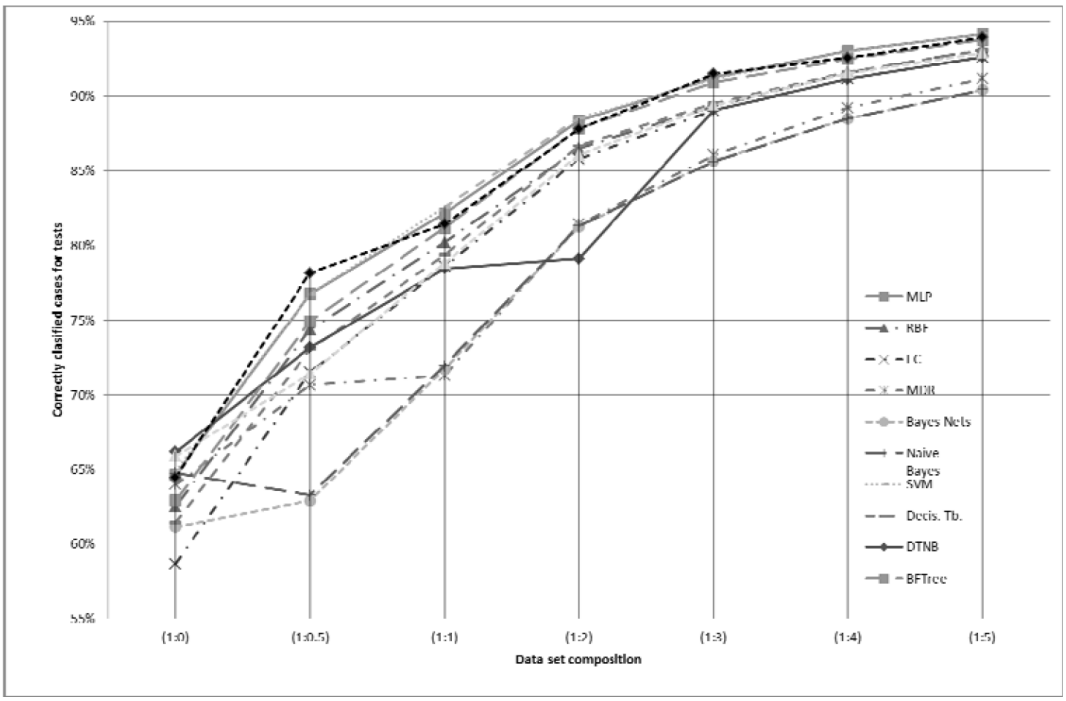


Fig. (2). Classification results of the different methods.

Table 1. Comparison between the LNN and the Variable Selection Method Proposed

Method	Classification Scores	AUC-ROC
LNN	78.20%	0.8405
Variable selection + SVM	76.98%	0.824

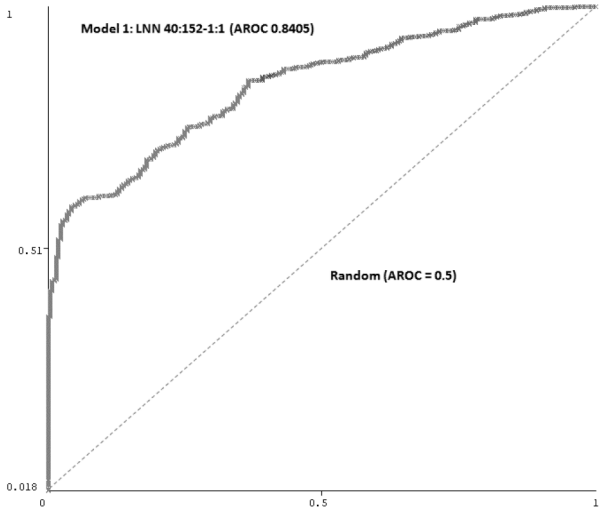


Fig. (3). Area under the receiver operating characteristic curve (AUC-ROC).

3.3. Single Nucleotide Polymorphism Schizophrenia Processing (SNP-Schizo)

Bio-AIMS (Biomedical Artificial Intelligence Model Server) [16, 23, 92, 93] is a portal that offers theoretical models based on Artificial Intelligence, Computational Biology and Bioinformatics to study Complex Systems in OMICS (Genomics, Transcriptomics, Metabolomics, Reactomics) that are relevant for Cancer, Neurosciences, Cardiovascular diseases, Parasitology, Microbiology and other Biomedical research in general (http://bio-aims.tic.udc.es/). It is the result of the collaboration between several scientific institutions. This portal includes two parts: TargetPred (Target Prediction) and DiseasePred (Disease Prediction). The DiseasePred part includes biomedicine applications for predicting human diseases from different data sources, such as genotypes. Future tools will be implemented based on the published models using EEG recordings and blood proteome mass spectra for epilepsy and colorectal cancer.

SNPSchizo (Single Nucleotide Polymorphism Schizophrenia Processing) [94] is the result of an online implementation (http://miaja.tic.udc.es/Bio-AIMS/SNPSchizo.php) of the previously described machine learning method which takes as input SNPs from two different genes related to schizophrenia and performs a classification [42] (see Fig. **4**). The interface of this tool was implemented using PHP, XHTML and Python, and the method was implemented using Java and Weka's [84] APIs. The tool is running on Apache HTTP – Server.

This tool is simple and easy to use. To get a classification result, the user has to introduce a list of sequences of SNPs in the format used by the tool and click on the "Diagnose" button. A new window will pop up with information about the results. These results can be saved as a text file and include the following information:

- For each sequence: the classification result (genetically predisposed to schizophrenia or not) and the SNP sequence.
- Information about the original dataset.
- Information about the method implemented online and its test accuracy.
- Input SNP order.
- Reference to the article of the comparative study with these data.

To test this tool, three example sequences are provided following the coding described above.

4. CONCLUSIONS

This review is presenting the applied computational techniques on schizophrenia, focusing on the genotype – disease relationships based on information of the nucleic acids such as the genetic mutations. Several machine learning methods have been described including a method for variable selection based on an ANN and a GA. To test the different methods, real clinical data of an association study on Galician patients (Spain) who suffered from schizophrenia using the DRD3 and HTR2A genes have been used, as well as simulated data which were generated with specialized software.

In complex diseases such as schizophrenia, the factors involved in increasing the risk of developing a disease do not correspond to one or two genes. There is a combination of values from different sets of SNPs, as well as a great influence due to environmental factors, which increase the risk of developing this complex disease.

The classification results obtained with the original data are not good with any of the presented methods. When the number of control subjects in the training sets is increased using simulated data, the developed method improves its classification accuracy, obtaining better results than with those methods which provide objective information about SNPs, obtaining a model based on rules or on trees. One of the models that obtain the best classification scores was implemented online as a free web tool named SNP-Schizo. The model implemented was the result of applying a LNN to the dataset that contained the lowest number of simulated subjects.

It is also interesting to observe which variables (or SNPs) are taken into account by the different methods which perform variable selection. Costas *et al.* used a sliding window approach and confirmed the existence of a common protective haplotype, which included the SNPs rs963468, rs2134655, rs1486012 and rs7631540 at DRD3, against schizophrenia [95]. The ANN and GA for variable selection approach presented in this paper is capable of finding three of the four previous SNPs (rs2134655, rs1486012 and rs7631540) if 17 variables are considered. However, if 21

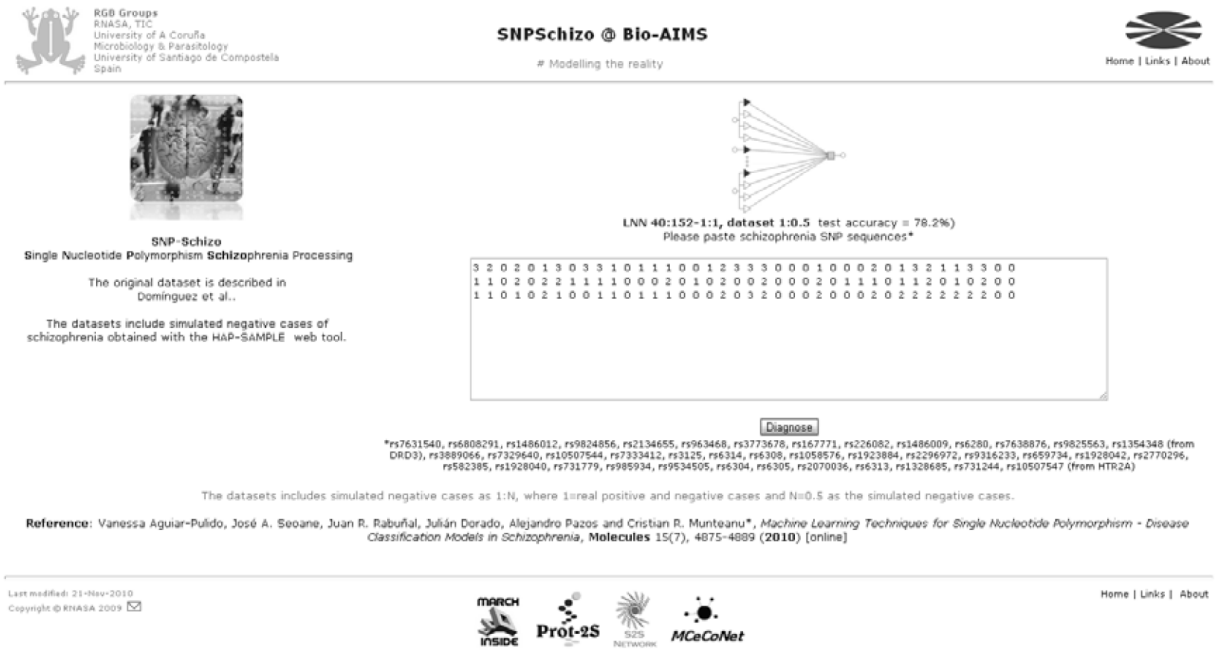


Fig. (4). SNP-Schizo web tool.

variables are considered, this approach finds all the SNPs included in the publication, as well as rs9824856, which is located at the same region. The same results in terms of classification scores and AUC-ROC values are obtained considering either 17 or 21 variables.

This review demonstrated the power of machine learning in obtaining genotype – disease classifications using molecular structure information such as the genetic mutations and it proposes the application on other diseases. The results can be used to determine the future gene targets for new drugs or genetic treatments.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

Vanessa Aguiar-Pulido and Cristian R. Munteanu acknowledge the funding support for a research position by the "Plan I2C" and "Isidro Parga Pondal" programs from Xunta de Galicia (Spain) and the European Social Fund (ESF), respectively, being the former one also co-funded by FEDER. This work is supported by the following projects: "Ibero-American Network of the Nano-Bio-Info-Cogno Convergent Technologies", Ibero-NBIC Network (209RT-0366) funded by CYTED (Spain), "Development of new image analysis techniques in 2D Gel for biomedical research " (ref.10SIN 105004PR) funded by Xunta de Galicia, RD07/0067/0005 funded by the Carlos III Health Institute, and "Galician Network for Colorectal Cancer Research" (REGICC, Ref. 2009/58), from the General Directorate of Scientific and Technologic Promotion of the Galician University System of Xunta de Galicia.

REFERENCES

[1] Picchioni, M.M.; Murray, R.M., Schizophrenia. *BMJ (Clinical research ed,* **2007**, *335*, (7610), 91-95.

[2] Svrakic, D.M.; Zorumski, C.F.; Svrakic, N.M.; Zwir, I.; Cloninger, C.R., Risk architecture of schizophrenia: the role of epigenetics. *Current opinion in psychiatry,* **2013**, *[Epub ahead of print]*.

[3] Devillers, J.; Balaban, A.T. *Topological Indices and Related Descriptors in QSAR and QSPR*. Gordon and Breach: The Netherlands, **1999**.

[4] Barabasi, A.L.; Bonabeau, E., Scale-free networks. *Scientific American,* **2003**, *288*, (5), 60-69.

[5] Balaban, A.T.; Basak, S.C.; Beteringhe, A.; Mills, D.; Supuran, C.T., QSAR study using topological indices for inhibition of carbonic anhydrase II by sulfanilamides and Schiff bases. *Molecular Diversity,* **2004**, *8*, (4), 401-412.

[6] Barabasi, A.L.; Oltvai, Z.N., Network biology: understanding the cell's functional organization. *Nature Reviews: Genetics,* **2004**, *5*, (2), 101-113.

[7] Barabasi, A.L., Sociology. Network theory-the emergence of the creative enterprise. *Science,* **2005**, *308*, (5722), 639-641.

[8] González-Díaz, H.; Vilar, S.; Santana, L.; Uriarte, E., Medicinal Chemistry and Bioinformatics – Current Trends in Drugs Discovery with Networks Topological Indices. *Current Topics in Medical Chemistry,* **2007**, *7*, (10), 1025-1039.

[9] Ferino, G.; Gonzalez-Diaz, H.; Delogu, G.; Podda, G.; Uriarte, E., Using spectral moments of spiral networks based on PSA/mass spectra outcomes to derive quantitative proteome-disease relationships (QPDRs) and predicting prostate cancer. *Biochemical and biophysical research communications,* **2008**, *372*, (2), 320-325.

[10] Gonzalez-Diaz, H.; Gonzalez-Diaz, Y.; Santana, L.; Ubeira, F.M.; Uriarte, E., Proteomics, networks and connectivity indices. *Proteomics,* **2008**, *8*, (4), 750-778.

[11] Randić, M., Novel shape descriptors for molecular graphs. *J Chem Inf Comput Sci,* **2001**, *41*, (3), 607-613.

[12] Balaban, A.T., Chemical graphs. XXXIV. Five new topological indices for the branching of tree-like graphs. *Theor. Chim. Acta,* **1979**, *53*, 355-375.

[13] Balaban, A.T.; Balaban, T.S., New vertex invariants and topological indices of chemical graphs based on information on distances. *J. Math. Chem.,* **1991**, *8*, 383-397.

[14] Gonzalez-Diaz, H.; Perez-Montoto, L.G.; Duardo-Sanchez, A.; Paniagua, E.; Vazquez-Prieto, S.; Vilas, R.; Dea-Ayuela, M.A.; Bolas-Fernandez, F.; Munteanu, C.R.; Dorado, J.; Costas, J.;

Ubeira, F.M., Generalized lattice graphs for 2D-visualization of biological information. *Journal of theoretical biology,* **2009**, *261*, (1), 136-147.

[15] González-Díaz, H.; Ferino, G.; Prado-Prado, F.J.; Vilar, S.; Uriarte, E.; Pazos, A.; Munteanu, C.R. In *An Omics Perspective on Cancer Research*. Cho, W.C.S., Ed.; Springer, **2010**.

[16] Munteanu, C.R.; Vazquez, J.M.; Dorado, J.; Sierra, A.P.; Sanchez-Gonzalez, A.; Prado-Prado, F.J.; Gonzalez-Diaz, H., Complex network spectral moments for ATCUN motif DNA cleavage: first predictive study on proteins of human pathogen parasites. *Journal of proteome research,* **2009**, *8*, (11), 5219-5228.

[17] Vazquez-Naya, J.M.; Martinez-Romero, M.; Porto-Pazos, A.B.; Novoa, F.; Valladares-Ayerbes, M.; Pereira, J.; Munteanu, C.R.; Dorado, J., Ontologies of drug discovery and design for neurology, cardiology and oncology. *Curr Pharm Des,* **2010**, *16*, (24), 2724-2736.

[18] Martinez-Romero, M.; Vazquez-Naya, J.M.; Rabunal, J.R.; Pita-Fernandez, S.; Macenlle, R.; Castro-Alvarino, J.; Lopez-Roses, L.; Ulla, J.L.; Martinez-Calvo, A.V.; Vazquez, S.; Pereira, J.; Porto-Pazos, A.B.; Dorado, J.; Pazos, A.; Munteanu, C.R., Artificial intelligence techniques for colorectal cancer drug metabolism: ontology and complex network. *Current drug metabolism,* **2010**, *11*, (4), 347-368.

[19] Munteanu, C.R.; Magalhaes, A.L.; Uriarte, E.; Gonzalez-Diaz, H., Multi-target QPDR classification model for human breast and colon cancer-related proteins using star graph topological indices. *Journal of theoretical biology,* **2009**, *257*, (2), 303-311.

[20] Martínez-Romero, M.; Vázquez-Naya, J.M.; Rabuñal, J.R.; Pita-Fernández, S.; Macenlle, R.; Castro-Alvariño, J.; López-Roses, L.; Ulla, J.L.; Martínez-Calvo, A.V.; Vázquez, S.; Pereira, J.; Porto-Pazos, A.B.; Dorado, J.; Pazos, A.; Munteanu, C.R., Artificial Intelligence Techniques for Colorectal Cancer Drug Metabolism: Ontologies and Complex Networks. *Current drug metabolism,* **2010**, *11*, (4), 347-368.

[21] Cruz-Monteagudo, M.; Munteanu, C.R.; Borges, F.; Cordeiro, M.N.; Uriarte, E.; Gonzalez-Diaz, H., Quantitative Proteome-Property Relationships (QPPRs). Part 1: finding biomarkers of organic drugs with mean Markov connectivity indices of spiral networks of blood mass spectra. *Bioorganic & medicinal chemistry,* **2008**, *16*, (22), 9684-9693.

[22] Gonzalez-Diaz, H.; Muino, L.; Anadon, A.M.; Romaris, F.; Prado-Prado, F.J.; Munteanu, C.R.; Dorado, J.; Sierra, A.P.; Mezo, M.; Gonzalez-Warleta, M.; Garate, T.; Ubeira, F.M., MISS-Prot: web server for self/non-self discrimination of protein residue networks in parasites; theory and experiments in Fasciola peptides and Anisakis allergens. *Mol Biosyst,* **2011**, *7*, (6), 1938-1955.

[23] Rodriguez-Soca, Y.; Munteanu, C.R.; Dorado, J.; Pazos, A.; Prado-Prado, F.J.; Gonzalez-Diaz, H., Trypano-PPI: a web server for prediction of unique targets in trypanosome proteome by using electrostatic parameters of protein-protein interactions. *Journal of proteome research,* **2010**, *9*, (2), 1182-1190.

[24] Rodriguez-Soca, Y.; Munteanu, C.R.; Dorado, J.; Rabuñal, J.; Pazos, A.; González-Díaz, H., Plasmod-PPI: a web-server predicting complex biopolymer targets in Plasmodium with entropy measures of protein-protein interactions. *Polymer,* **2010**, *51*, (1), 264-273.

[25] den Dunnen, J.T.; Antonarakis, S.E., Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion. *Human mutation,* **2000**, *15*, (1), 7-12.

[26] Ban, H.J.; Heo, J.Y.; Oh, K.S.; Park, K.J., Identification of type 2 diabetes-associated combination of SNPs using support vector machine. *BMC genetics,* **2010**, *11*, 26.

[27] Saangyong Uhmn, D.-H.K., Young-Woong Ko, Sungwon Cho, Jaeyoun Cheong and Jin Kim, A study on application of single nucleotide polymorphism and machine learning techniques to diagnosis of chronic hepatitis. *Expert Systems,* **2009**, *26*, (1), 10.

[28] Briggs, F.B.; Ramsay, P.P.; Madden, E.; Norris, J.M.; Holers, V.M.; Mikuls, T.R.; Sokka, T.; Seldin, M.F.; Gregersen, P.K.; Criswell, L.A.; Barcellos, L.F., Supervised machine learning and logistic regression identifies novel epistatic risk factors with PTPN22 for rheumatoid arthritis. *Genes and immunity,* **2010**, *11*, (3), 199-208.

[29] Nicodemus, K.K.; Callicott, J.H.; Higier, R.G.; Luna, A.; Nixon, D.C.; Lipska, B.K.; Vakkalanka, R.; Giegling, I.; Rujescu, D.; Clair, D.S.; Muglia, P.; Shugart, Y.Y.; Weinberger, D.R., Evidence of statistical epistasis between DISC1, CIT and NDEL1 impacting risk for schizophrenia: biological validation with functional neuroimaging. *Human genetics*.

[30] Lu, A.T.; Bakker, S.; Janson, E.; Cichon, S.; Cantor, R.M.; Ophoff, R.A., Prediction of serotonin transporter promoter polymorphism genotypes from single nucleotide polymorphism arrays using machine learning methods. *Psychiatric genetics,* **2012**, *22*, (4), 182-188.

[31] Meltzer, H.Y.; Matsubara, S.; Lee, J.C., Classification of typical and atypical antipsychotic drugs on the basis of dopamine D-1, D-2 and serotonin2 pKi values. *The Journal of pharmacology and experimental therapeutics,* **1989**, *251*, (1), 238-246.

[32] Suzuki, M.; Hurd, Y.L.; Sokoloff, P.; Schwartz, J.C.; Sedvall, G., D3 dopamine receptor mRNA is widely expressed in the human brain. *Brain research,* **1998**, *779*, (1-2), 58-74.

[33] Dominguez, E.; Loza, M.I.; Padin, F.; Gesteira, A.; Paz, E.; Paramo, M.; Brenlla, J.; Pumar, E.; Iglesias, F.; Cibeira, A.; Castro, M.; Caruncho, H.; Carracedo, A.; Costas, J., Extensive linkage disequilibrium mapping at HTR2A and DRD3 for schizophrenia susceptibility genes in the Galician population. *Schizophrenia research,* **2007**, *90*, (1-3), 123-129.

[34] Diederich, J. *Artificial neural networks: concept learning*. IEEE Press Piscataway, NJ, USA, **1990**.

[35] Byvatov, E.; Schneider, G., Support vector machine applications in bioinformatics. *Applied Bioinformatics,* **2003**, *2*, (2), 67-77.

[36] Byvatov, E.; Schneider, G., Support vector machine applications in bioinformatics. *Appl Bioinformatics,* **2003**, *2*, (2), 67-77.

[37] Cristianini, N.; Shawe-Taylor, J. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, USA, **1999**.

[38] Gomez-Carracedo, M.P.; Gestal, M.; Dorado, J.; Andrade, J.M., Chemically driven variable selection by focused multimodal genetic algorithms in mid-IR spectra. *Analytical and Bioanalytical Chemistry,* **2007**, *389*, (7-8), 2331-2342.

[39] Gestal, M.; Gómez-Carracedo, M.P.; Andrade, J.M.; Dorado, J.; Fernández, E.; Prada, D.; Pazos, A., Selection of variables by genetic algorithms to classify apple beverages by artificial neural networks. *Applied Artificial Intelligence,* **2005**, *19*, (2), 181-198.

[40] Aguiar Pulido, V.; Seoane Fernández, J.A.; Freire, A.; Munteanu, C.R., Data Mining in Complex Diseases Using Evolutionary Computation. *Lecture Notes in Computer Science,* **2009**, *5517*, (Part I), 917-924.

[41] Tan, P.-N.; Steinbach, M.; Kumar, V. *Introduction to Data Mining*. Pearson Addition Wesley: Boston, Maryland, **2006**.

[42] Aguiar-Pulido, V.; Seoane, J.A.; Rabunal, J.R.; Dorado, J.; Pazos, A.; Munteanu, C.R., Machine learning techniques for single nucleotide polymorphism - disease classification models in schizophrenia. *Molecules (Basel, Switzerland),* **2010**, *15*, (7), 4875-4889.

[43] Wright, F.A.; Huang, H.; Guan, X.; Gamiel, K.; Jeffries, C.; Barry, W.T.; de Villena, F.P.; Sullivan, P.F.; Wilhelmsen, K.C.; Zou, F., Simulating association studies: a data-based resampling method for candidate regions or whole genome scans. *Bioinformatics,* **2007**, *23*, (19), 2581-2588.

[44] Kingman, J.F., Origins of the coalescent. 1974-1982. *Genetics,* **2000**, *156*, (4), 1461-1463.

[45] Liang, L.; Zollner, S.; Abecasis, G.R., GENOME: a rapid coalescent-based whole genome simulator. *Bioinformatics,* **2007**, *23*, (12), 1565-1567.

[46] Balloux, F., EASYPOP (version 1.7): a computer program for population genetics simulations. *The Journal of heredity,* **2001**, *92*, (3), 301-302.

[47] Hey, J. A computer program for forward population genetic simulation, 2004.

[48] Hoggart, C.J.; Chadeau-Hyam, M.; Clark, T.G.; Lampariello, R.; Whittaker, J.C.; De Iorio, M.; Balding, D.J., Sequence-level population simulations over large genomic regions. *Genetics,* **2007**, *177*, (3), 1725-1731.

[49] Peng, B.; Kimmel, M., simuPOP: a forward-time population genetics simulation environment. *Bioinformatics,* **2005**, *21*, (18), 3686-3687.

[50] Edwards, T.L.; Bush, W.S.; Turner, S.D.; Dudek, S.M.; Torstenson, E.S.; Schmidt, M.; Martin, E.; Ritchie, M.D., Generating Linkage Disequilibrium Patterns in Data Simulations using genomeSIMLA. *LNCS: Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics,* **2008**, *4973*, (2008), 24-35.

[51] Li, J.; Chen, Y., Generating samples for association studies based on HapMap data. *BMC bioinformatics,* **2008**, *9*, 44.

[52] Bishop, C. *Neural Networks for pattern recognition*. Oxford University Press New York, **1995**.

[53] Haykin, S.; Network, N., A comprehensive foundation. *Neural Networks,* **2004**, *2*.

[54] Zhang, W. *Computational Ecology: Artificial Neural Networks and Their Applications*. World Scientific Publishing Company Incorporated, **2010**.

[55] Vapnik, V. *Statistical Learning Theory*. John Weily and Sons: New York, **1998**.

[56] John, G.H.; Langley, P. In *Estimating Continuous Distributions in Bayesian Classifiers*, Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, San Mateo; Morgan Kaufmann, **1995**; pp 338-345.

[57] Bouckaert, R.R. *Bayesian networks in Weka*; Computer Science Department. University of Waikato: 2004.

[58] Kohavi, R. In *The Power of Decision Tables*, Proceedings of 8th European Conference on Machine Learning, Heraclion, Greece; Springer-Verlag, **1995**; pp 174-189.

[59] Hall, M.; Frank, E. In *21st Florida Artificial Intelligence Society Conference*; AAAI Press: Florida, USA, **2008**, pp 318-.319.

[60] Shi, H. *Best-first decision tree learning*. Hamilton: New Zeland, **2007**.

[61] Yoav Freund, R.E.S. In *Experiments with a new boosting algorithm*, Thirteenth International Conference on Machine Learning, San Francisco, **1996**.

[62] Moore, J.H.; Gilbert, J.C.; Tsai, C.T.; Chiang, F.T.; Holden, T.; Barney, N.; White, B.C., A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. *Journal of theoretical biology,* **2006**, *241*, (2), 252-261.

[63] Cordell, H.J., Detecting gene-gene interactions that underlie human diseases. *Nature reviews: genetics,* **2009**, *10*, (6), 392-404.

[64] Greene, C.S.; Sinnott-Armstrong, N.A.; Himmelstein, D.S.; Park, P.J.; Moore, J.H.; Harris, B.T., Multifactor dimensionality reduction for graphics processing units enables genome-wide testing of epistasis in sporadic ALS. *Bioinformatics,* **2010**, *26*, (5), 694-695.

[65] Cattaert, T.; Urrea, V.; Naj, A.C.; De Lobel, L.; De Wit, V.; Fu, M.; Mahachie John, J.M.; Shen, H.; Calle, M.L.; Ritchie, M.D.; Edwards, T.L.; Van Steen, K., FAM-MDR: a flexible family-based multifactor dimensionality reduction technique to detect epistasis using related individuals. *PloS one*, *5*, (4), e10304.

[66] He, H.; Oetting, W.S.; Brott, M.J.; Basu, S., Pair-wise multifactor dimensionality reduction method to detect gene-gene interactions in a case-control study. *Human heredity,* **2009**, *69*, (1), 60-70.

[67] Kang, S.G.; Lee, H.J.; Choi, J.E.; Park, Y.M.; Park, J.H.; Han, C.; Kim, Y.K.; Kim, S.H.; Lee, M.S.; Joe, S.H.; Jung, I.K.; Kim, L., Association Study between Antipsychotics - Induced Restless Legs Syndrome and Polymorphisms of Dopamine D1, D2, D3, and D4 Receptor Genes in Schizophrenia. *Neuropsychobiology,* **2008**, *57*, (1-2), 49-54.

[68] Vilella, E.; Costas, J.; Sanjuan, J.; Guitart, M.; De Diego, Y.; Carracedo, A.; Martorell, L.; Valero, J.; Labad, A.; De Frutos, R.; Najera, C.; Molto, M.D.; Toirac, I.; Guillamat, R.; Brunet, A.; Valles, V.; Perez, L.; Leon, M.; de Fonseca, F.R.; Phillips, C.; Torres, M., Association of schizophrenia with DTNBP1 but not with DAO, DAOA, NRG1 and RGS4 nor their genetic interaction. *Journal of psychiatric research,* **2008**, *42*, (4), 278-288.

[69] Yasuno, K.; Ando, S.; Misumi, S.; Makino, S.; Kulski, J.K.; Muratake, T.; Kaneko, N.; Amagane, H.; Someya, T.; Inoko, H.; Suga, H.; Kanemoto, K.; Tamiya, G., Synergistic association of mitochondrial uncoupling protein (UCP) genes with schizophrenia. *American Journal of Medical Genetics. Neuropsychiatric Genetics,* **2007**, *144B*, (2), 250-253.

[70] Holland, J. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, **1975**.

[71] Gestal, M.; Rivero, D.; Rabuñal, J.R.; Dorado, J.; Pazos, A. *Introducción a los Algoritmos Genéticos y Programación Genética*. Servicio de Publicaciones Universidade da Coruña, **2010**.

[72] Hernández, J.A.; Dorado, J.; Gestal, M.; Porto, A.B., Avances en Algoritmos Evolutivos. *INTELIGENCIA ARTIFICIAL Y COMPUTACIÓN AVANZADA*, 35.

[73] Darwin, C. On the Origin of Species by Means of Natural Selection. John Murray, **1859**.

[74] Guo, L.; Rivero, D.; Dorado, J.; Munteanu, C.R.; Pazos, A., Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications,* **2011**, *38*, (8), 10425-10436.

[75] Rivero, D.; Dorado, J.; Rabuñal, J.R.; Pazos, A., Modifying genetic programming for artificial neural network development for data mining. *Soft Computing-A Fusion of Foundations, Methodologies and Applications,* **2009**, *13*, (3), 291-305.

[76] Rivero, D.; Dorado, J.; Rabuñal, J.R.; Pazos, A.; Pereira, J., Artificial neural network development by means of genetic programming with graph codification. *Transactions on Engineering, Computing and Technology,* **2006**, *16*, 209-214.

[77] Cantú-Paz, E. In *Genetic and Evolutionary Computation Conference*; Springer-Verlag: Seattle, Washington, USA, **2004**, pp 959–970.

[78] Fidelis, M.V.; Lopes, H.; Freitas, A. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*; IEEE, **2000**; Vol. *1*, pp 805-810.

[79] Dybowski, R.; Gant, V.; Weller, P.; Chang, R., Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm. *The Lancet,* **1996**, *347*, (9009), 1146-1150.

[80] Anbarasi, M.; Anupriya, E.; Iyengar, N., Enhanced prediction of heart disease with feature subset selection using genetic algorithm. *International Journal of Engineering Science and Technology,* **2010**, *2*, (10), 5370-5376.

[81] Gestal, M.; Andrade, J.M. In *Encyclopedia of Artificial Intelligence*; Information Science Reference: Hershey, EEUU, **2008**, pp 581-588.

[82] Seoane, J.; Aguiar, V.; Gestal, M.; Dorado, J.; Pazos, A. In *Association analysis in complex diseases using evolutionary computation*, 6th Intelligent System for Molecular Biology, Toronto, Canada, **2008**; p 22.

[83] Gestal, M.; Cancela, A.; Andrade, J.; Gomez-Carracedo, M. In *Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications*. Sugaraman, V., Ed.; Information Science Reference: Hershey, New York, USA, **2007**, pp 274-292.

[84] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.A., The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* **2009**, *11*, (1).

[85] Russel, S.; Norvig, P. *Artificial Intelligence: A Modern Approach (2nd Ed.)*. Prentice Hall: Upper Saddle River, New Jersey: Prentice Hall, **2003**.

[86] Gutlein, M.; Frank, E.; Hall, M.; Karwath, A. In *In Proceedings of Symposium on Computational Intelligence and Data Mining*; IEEE Computer Society: Nashville, TN, **2009**, pp 332-339.

[87] Yu, L.; Liu, H. In Proceedings of the Twentieth International Conference on Machine Learning, **2003**, pp 856-863.

[88] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co.: Boston, MA, **1989**.

[89] Garcia Lopez, F.; Garcia Torres, M.; Melian Batista, B.; Moreno Perez, J.A.; Moreno-Vega, J.M., Solving feature subset selection problem by a Parallel Scatter Search. *European Journal of Operational Research,* **2006**, *169*, (2), 477-489.

[90] Liu, H.; Setiono, R. In *13th International Conference on Machine Learning*: Bari, Italy, **1996**, pp 319-327.

[91] Landwehr, N.; Hall, M.; Frank, E., Logistic model trees. *Machine Learning,* **2005**, *59*, (1), 161-205.

[92] Gonzalez-Diaz, H.; Prado-Prado, F.J.; Garcia-Mera, X.; Alonso, N.; Abeijon, P.; Caamano, O.; Yanez, M.; Munteanu, C.R.; Pazos Sierra, A.; Dea-Ayuela, M.A.; Gomez-Munoz, M.T.; Garijo, M.M.; Sansano, J.; Ubeira, F.M., MIND-BEST: web server for drugs & target discovery; design, synthesis, and assay of MAO-B inhibitors and theoretic-experimental study of G3PD protein from Trichomona gallineae. *Journal of proteome research,* **2010**.

[93] Concu, R.; Dea-Ayuela, M.A.; Perez-Montoto, L.G.; Prado-Prado, F.J.; Uriarte, E.; Bolas-Fernandez, F.; Podda, G.; Pazos, A.; Munteanu, C.R.; Ubeira, F.M.; Gonzalez-Diaz, H., 3D entropy and moments prediction of enzyme classes and experimental-theoretic study of peptide fingerprints in Leishmania parasites. *Biochimica et biophysica acta,* **2009**, *1794*, (12), 1784-1794.

[94] Aguiar-Pulido, V.; Seoane, J.; Munteanu, C.; Pazos, A., SNP-Schizo: A Web Tool for Schizophrenia SNP Sequence

Classification. *LNCS: Advances in Computational Intelligence,* **2011**, *6692*, 252-259.

[95] Costas, J.; Carrera, N.; Dominguez, E.; Vilella, E.; Martorell, L.; Valero, J.; Gutierrez-Zotes, A.; Labad, A.; Carracedo, A., A common haplotype of DRD3 affected by recent positive selection is associated with protection from schizophrenia. *Human genetics,* **2009**, *124*, (6), 607-613.

Received: January 31, 2013 **Revised: February 26, 2013** **Accepted: March 09, 2013**

Índice de términos