UNIVERSIDAD CENTRAL "Marta Abreu" de Las Villas.

Facultad Matemática, Física y Computación

Trabajo para optar por el Título de

Licenciado en Ciencia de la Computación

Título

"Descubrimiento de Descriptores Moleculares Óptimos Mediante Computación Evolutiva"

AUTOR:

Yasser Silveira Vaz d'Almeida

TUTOR:

Lic. José Ricardo Valdés Martini

Dr. Yovani Marrero Ponce

Santa Clara, 2011

Dictamen con Derechos de Autor cedidos a MFC

El que suscribe, Yasser Silveira Vaz d'Almeida, hago constar que el trabajo titulado por "Descubrimiento de Descriptores Moleculares Óptimos Mediante Computación Evolutiva" fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de los estudios de la especialidad en Licenciatura en Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

_____

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

Firma del tutor

_____

_____

Firma del jefe del Laboratorio

_____

Dedicatoria

Dedico este trabajo a mi padre Armindo Vaz d'Almeida, mi madre Maria da Conceição Silveira d'Almeida, mis hermanas Nayda Indhira Silveira d'Almeida y Niurka Solange Silveira d'Almeida y Anisa Rodrigues Afonso quienes siempre han estado a mi lado y me han dado la fuerza necesaria para ultrapasar todos los obstáculos aunque cuando pensaba que no los podía ultrapasar.

Por ultimo una dedicatoria especial a mi hija Lidia Sophia Rodrigues Afonso Silveira d'Almeida quien me ha dado mucha suerte.

Agradecimientos

Agradezco la Revolución Cubana por darme la oportunidad de cursar mi carrera, a todo el colectivo de profesores quienes me han ayudado y enseñado, especialmente al profesor Doctor José Talavera quien me ha ayudado a tener una excelente base de matemática, al Profesor Roman Figueras quien me ha ayudado a programar, mis tutores Dr Yovani Marrero Ponce, Lic. José Ricardo Martiní, Dr. Carlos Morell, Lic. Stephen Jones Barigye quien me ha ayudado mucho con la parte de química y farmacia para realizar este trabajo. Mis amigos Davidson Narciso, Gelson Nunes, Saminio Lima, Evy Mette, Dhana Lazarus, Yuri Sebastião, Ebenny Pinheiro quienes han estado siempre a mi lado y apoyado.

Resumen

La industria farmacéutica necesita tratar con el aumento del coste y el tiempo requerido para el desarrollo de nuevos fármacos, y el descubrimiento “in silico” y posterior optimización de compuestos líderes se está convirtiendo en un medio cada vez más importante para alcanzar este objetivo.

Este proceso implica, a menudo, la obtención de modelos para estimar las relaciones cuantitativas de estructura-actividad, que se centran en predecir la actividad biológica de un compuesto a partir de una representación vectorial de la estructura molecular. Las componentes de esta representación vectorial de la estrutura molecular no son más que descriptores moleculares.

Este trabajo consiste en diseñar e implementar una herramienta computacional que permita la obtención de modelos QSAR de predicción óptima para la descripción de propiedades químico-físicas y biológicas de compuestos orgánicos empleando la exploración del espacio de todos los posibles descriptores moleculares algebraicos TOMOCOMD-CARDD, mediante el uso de algoritmos genéticos que encontraran sus parámetros idóneos.

Palabras Clave: Modelos QSAR, TOMOCOMD-CARDD, Algoritmos Genéticos, Aprendizaje Automatizado.

Abstract

The pharmaceutical industry needs to deal with the risising cost of and the time required for the development of new drugs, and the ín silico`discovery and later optimization of compound leaders is being converted into a medium each time more important to meet this objective.

This process often implies the obtention of models to estimate the qualitative relations of structure - activity, that are centred on predicting the biological activity of a compound beginning from a vectoral of a molecular structure. The components of this vectoral representation of the molecular structure are no more than molecular descriptors.

This investigation consists of the design and implementation of a computational tool for the obtention of QSAR models of optimum prediction in the description of chemical – physical and biological properties of organic compounds employing the exploration of the space of all the possible algebraic molecular descriptors TOMOCOMD-CARDD, using genetic algorithms to find the best parameters.

Key Words: QSAR Models, TOMOCOMD-CARDD, Genetic Algorithms, Machine Learning.

Tabla de Contenidos

Introducción

La industria farmacéutica necesita tratar con el aumento del coste y el tiempo requerido para el desarrollo nuevos fármacos, y el descubrimiento "in silico" y posterior optimización de compuestos líderes se está convirtiendo en un medio cada vez más importante para alcanzar este objetivo. (Charton 1996)

Este proceso implica, a menudo, la obtención de modelos para estimar las relaciones cuantitativas de estructura-actividad (Diudea 2001), que se centran en predecir la actividad biológica de un compuesto a partir de una representación vectorial de la estructura molecular.(Louis 2003),(Kniaz 2000) Además de los estudios QSAR referidos a la descripción de la actividad, los estudios QSPR/QSTR (siglas en inglés acrónimos de Quantitative Structure Property/Toxicity Relationships) también se han convertido en una importante área de investigación en la química computacional. (A Katrizky 2000)

Este tipo de estudios se encuentra en la intersección entre la biología, la química y la computación y tienen dos objetivos fundamentales. El primero es brindar una vía para estimar, con un aceptable grado de precisión, la actividad/propiedad/toxicidad estudiada a nuevos compuestos. El segundo, pero no menos importante, es obtener una interpretación en términos estructurales de la actividad/propiedad/toxicidad estudiada. El paradigma elaborado en los estudios QSAR/QSPR/QSTR (en lo adelante se utilizará solo el término QSAR) está relacionado con el hecho de que las propiedades físicas, físico-químicas, químicas, biológicas y toxicológicas de los compuestos orgánicos dependen en último término de la estructura molecular. (Singh 2000)

Los índices o descriptores moleculares (DM) representan la vía por la cual la estructura química se transforma en números permitiendo el tratamiento matemático de la información química contenida en la molécula y pueden definirse como representaciones matemáticas de las moléculas que se obtienen al aplicar algoritmos específicos sobre una representación molecular definida o a partir de procedimientos experimentales específicos. (Karelson 2000) and (van de Waterbeemd 1998)

La naturaleza de los descriptores, depende de cual haya sido el proceder utilizado para la definición de los mismos, pudiendo tener en cuenta rasgos topológicos (Marrero-Ponce and Torrens 2006), geométricos (Estrada and Molina 2001), y electrónicos de las moléculas. Algunos

de estos descriptores sin embargo, tienen "más información" de propiedades físico-químicas que de los rasgos estructurales de la molécula. Estos incluyen los basados en la determinación experimental de propiedades físico-químicas, tales como la mayoría de las constantes de los sustituyentes, hidrofobias, electrónicas y estéricas. (Kubinyi 1993) En contraste, los llamados **índices topológicos** (Todeschini and Consonni 2000) *tienen la información estructural contenida en una representación bidimensional de las moléculas, generalmente el grafo molecular con los átomos de hidrógenos suprimidos, sin considerar ningún rasgo físico-químico de las moléculas*. La mayoría de estos índices pueden considerarse como descriptores estructurales explícitos. Otro grupo de descriptores, llamados químico-cuánticos describen rasgos electrónicos de las moléculas basados en el uso de la función de onda molecular. Los descriptores geométricos tienen información de los rasgos estructurales 3D de las moléculas en una vía explícita, tales como distancia y ángulos de enlaces o en una vía implícita, en forma de descriptores topográficos. Los IT han comenzado a ocupar un lugar importante dentro del conjunto de descriptores moleculares utilizados en los estudios QSAR, siendo probablemente el diseño/descubrimiento de nuevos compuestos bioactivos, una de las más activas áreas de investigación donde se aplican estos descriptores a problemas biológicos.

Teniendo en cuenta lo planteado anteriormente, fue definida recientemente tres nuevas familia de índices topológicos (topo-químico) a partir de la aplicación de conceptos de la matemática discreta y el álgebra lineal a la química. Estos descriptores están basados en el cálculo de formas cuadráticas, lineales y bilineales y han sido aplicados en diversos estudios QSAR/QSPR obteniéndose resultados satisfactorios. Sin embargo, no siempre estos índices muestran un desempeño totalmente satisfactorio para la predicción de ciertas propiedades. De hecho no se puede esperar que un conjunto específico de índices sea superior absolutamente a otros conjuntos posibles y/o pueda producir buenos resultados en todos los problemas. Como ejemplo de lo antes planteado tenemos el estudio realizado para predecir la permeabilidad de moléculas orgánicas y el trabajo referido a la predicción de estabilidad de péptidos y proteínas, entre otros.

Por otro lado, dado el gran número de índices algebraicos que pueden ser calculados a partir de la combinación total de las diferentes partes y/o parámetros de estos índices, la selección de atributos debe ser empleada con el propósito de identificar los rasgos más relacionados con la

propiedad de interés, para disminuir la colinealidad entre variables, disminuir la dimensión a la hora de construir los modelos finales usando método wrapper, etc.

Es decir, el **problema científico** de esta investigación asociado a esta área del saber, es la determinación de los descriptores moleculares idóneos (y su combinación óptima) a partir de un número grande de posibles índices para la predicción de una actividad específica es un problema actual del QSAR, sobre todo para datas de alta dimensión como las que se generan con el empleo de los índices TOMOCOMD-CARDD que están basados en formas algebraicas. El espacio de búsqueda es de tamaño exponencial directamente proporcional a la cantidad de descriptores empleados lo cual está directamente relacionado con la combinación total de las diferentes partes y/o parámetros de estos índices, las cuales en este proyecto se planean aumentar considerablemente al introducir extensiones y generalización en varias partes de las formas algebraicas, por lo que estos problemas no pueden ser resueltos por métodos de búsqueda exhaustivos.

Por ello como **hipótesis**, la definición e implementación computacionalmente de nuevas familias de índices moleculares basados en extensiones y generalizaciones de las formas algebraicas ya existentes, al igual que el empleo de los algoritmos genéticos (GOLDBERG 1989) como herramienta para explorar el espacio de los descriptores moleculares pudiera permitir encontrar modelos de predicción que se ajusten mejor a los datos y deben ayudar a resolver – o al menos resolver mejor – problemas de predicción de propiedades físicas, química, químico-físicas y biológicas de nuevos compuestos.

En este contexto el **objetivo general** de esta investigación consiste, en diseñar e implementar una GUI que permita la obtención de modelos de predicción óptimos para la descripción de propiedades químico-físicas y biológicas de compuestos orgánicos empleando una exploración del espacio de todos los posibles DM mediante el uso de algoritmos genéticos que encontraran los parámetros idóneos de los índices TOMOCOM-CARDD (Marrero-Ponce and Romero 2002) basados en formas algebraicas extendidas y generalizadas.

En la consecuente de este objetivo nos guían a las siguientes **preguntas de investigación**:

1. ¿Cómo definir nuevos descriptores moleculares de manera que amplíen los ya existentes considerando información topológica y química no incluida en los descriptores actuales?

2. ¿Qué herramienta computacional se debe crear de manera que permita calcular los distintos descriptores moleculares a un grupo diverso de moléculas?
3. ¿Cómo modelar el problema de selección de descriptores moleculares?
4. ¿Qué herramienta computacional se debe crear de manera tal que permita explorar el espacio de los descriptores moleculares para encontrar aquellos que permitan definir un modelo de predicción que se ajuste mejor a los datos?
5. ¿Cómo validar y verificar las herramientas computacionales?

Para lograr nuestro objetivo general nos hemos planteado los siguientes **objetivos específicos**:

1. Definir e implementar computacionalmente nuevos DM basados en la teoría de grafos y el álgebra lineal que amplíen los ya existentes considerando información topológica y química no incluida en los descriptores actuales.
2. Diseñar e implementar una GUI interfaz gráfica de usuario que permita calcular los distintos descriptores moleculares a un grupo diverso de moléculas.
3. Modelar el problema de selección de descriptores moleculares mediante algoritmos genéticos.
4. Diseñar e implementar una interfaz gráfica de usuario que permita explorar el espacio de los descriptores moleculares mediante algoritmos genéticos para encontrar las combinaciones óptimas que permitan definir modelos de predicción que mejor se ajuste a los datos.
5. Validar y verificar la herramienta mediante la utilización de conjuntos de datos internacionales.

En resumen de esta investigación podemos enunciar como **novedad científica**, que este trabajo está fundamentado en la definición de nuevas familias de ITs basados en la extensión y generalización de las formas algebraicas ya existentes utilizando nuevas propiedades atómicas para caracterizar los núcleos atómicos de la molécula, nuevos representaciones matriciales, nuevos índices locales, trucajes, etc. Los DMs totales y locales (para grupos de átomos o enlaces) se definen empleando una serie de invariantes de distancia (normas), medias y estadísticos, a partir de vectores conformados por ITs atómicos. Finalmente se utilizan, por primera vez, los

algoritmos genéticos para el descubrimiento de los índices más idóneos para la generación de modelos óptimos para la descripción de propiedades químico-físicas y biológicas.

Este trabajo se ha estruturado en tres capítulos. El primero capítulo está dedicado al marco teórico de nuestra investigación, o sea la Computación Evolutiva y la Química Computacional, específicamente sobre los Algoritmos Genéticos, las Estratégicas QSPR, AG aplicados a Química Computacional y por último sobre los Descriptores Moleculares. En él se analizan y exponen aquellas teorías, enfoques teóricos, investigaciones y antecedentes en general válidos para el correcto encuadre del estudio. Como resultado se adoptarán las herramientas necesarias para la consecución de los objetivos. El segundo está dedicado al desarrollo de nuevos DM, su implementación en la bilbioteca seleccionada, el diseño e implementación de la herramienta computacional para el calculo de estos DM para un conjunto de moléculas dado, la molelación del problema de selección de estos DM mediante AG, diseño e implementación de una herramienta computacional que permita explorar el espacio de estos DM mediante AG. El tercer capítulo se analiza los resultados de la validación y verificación de las herramientas mediante la utilación de conjuntos de datos internacionales. Al finalizar se proveen las conclusiones de nuestro trabajo así como propuestas para trabajos futuros.

1 Química Computacional

Este capítulo presentamos la Computación Evolutiva, en la Química Computacional, específicamente sobre los Algoritmos Genéticos, las Estratégicas QSPR, AG aplicados a Química Computacional y por último sobre los Descriptores Moleculares.

1.1 Algoritmos Genéticos

Expuesto concisamente, un algoritmo genético (GOLDBERG 1989) es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas y constituyen una de las técnicas de computación evolutiva más difundidas en la actualidad, como consecuencia de su versatilidad para resolver un amplio rango de problemas. Al constituir un caso de técnica evolutiva, los AG basan su operativa en una emulación de la evolución natural de los seres vivos, trabajando sobre una población de soluciones potenciales evoluciona de acuerdo a interacciones y transformaciones únicas. Los individuos que constituyen la población se esfuerzan por sobrevivir: una selección programada en el proceso evolutivo, inclinada hacia los individuos más aptos, determina aquellos individuos que formarán parte de la siguiente generación. El grado de adaptación de un individuo se evalúa de acuerdo al problema a resolver, mediante la definición de una función de adecuación al problema, la función de fitness. (Holland 1975), (GOLDBERG 1989), (Holland 1975), (MICHALEWICZ 1992)

Bajo ciertas condiciones, el mecanismo definido por los operadores inspirados por la genética natural y la evolución darwiniana lleva a la población a converger hacia una solución aproximada al óptimo del problema, luego de un determinado número de generaciones.

En su formulación clásica, los algoritmos genéticos se basan en el esquema genérico de un AG presentado en la Figura1.0. A partir de este esquema, el algoritmo genético define “Operadores Evolutivos” que implementan la recombinación de individuos (el operador de cruzamiento) y la variación aleatoria para proporcionar diversidad (el operador de mutación).

```
Inicializar(P(0))
generación=0
mientras (no CriterioParada) hacer
      Evaluar(P(generación))
      Padres = Seleccionar(P(generación))
      Hijos = Aplicar Recombinación (Padres)
      Hijos = Aplicar Mutación (Hijos)
      NuevaPob = Reemplazar(Hijos, P(generación))
      generación ++
      P(generación) = NuevaPob
fin
retornar Mejor Solución Encontrada
```

Figura 1.0: Esquema genérico de un algoritmo genético.

La característica distintiva de los algoritmos genéticos respecto a las otras técnicas evolutivas consiste en su uso fundamental del cruzamiento como operador principal, mientras que la mutación se utiliza como operador secundario tan solo para agregar una nueva fuente de diversidad en el mecanismo de exploración del espacio de soluciones del problema.

Inclusive la mutación puede llegar a ser un operador opcional o estar ausente en algunas variantes de algoritmos genéticos que utilizan otros operadores para introducir diversidad. (GOLDBERG 1989)

En general, los algoritmos genéticos se han utilizado para trabajar con codificaciones binarias para problemas de búsqueda en espacios de cardinalidad numerable, aunque su alto nivel de aplicabilidad ha llevado a proponer su trabajo con codificaciones reales, e inclusive con codificaciones no tradicionales, dependientes de los problemas a resolver.

Algunos autores consideran que el uso del mecanismo de selección denominado selección proporcional, que determina la cantidad de copias de individuos a considerar en la recombinación de forma proporcional a sus valores de fitness, es una característica que distingue a los algoritmos genéticos (BACK, FOGEL et al. 1997). Pero tomando en cuenta la diversidad de mecanismos de selección utilizados en las propuestas de algoritmos genéticos en los últimos años es posible concluir que si bien en general se recurre a la selección proporcional, otros mecanismos son igualmente utilizados para modificar el proceso de exploración del espacio de soluciones de los diferentes problemas abordados.

1.1.1 Representación de soluciones

Los algoritmos genéticos no trabajan directamente sobre las soluciones del problema en cuestión, sino que lo hacen sobre una abstracción de los objetos solución, usualmente

denominadas *cromosomas* por analogía con la evolución natural biológica. Un *cromosoma* es un vector de *genes*, mientras que el valor asignado a un *gen* se denomina *alelo*.

En la terminología biológica, *genotipo* denota al conjunto de *cromosomas* que definen las características de un *individuo*. El *genotipo* sometido al medio ambiente se denomina *fenotipo*. En términos de los algoritmos genéticos el *genotipo* también está constituido por *cromosomas*, utilizándose generalmente un único *cromosoma* por *individuo* solución al problema. Por ello suelen utilizarse indistintamente los términos *genotipo*, *cromosoma* e *individuo*. Por su parte, el *fenotipo* representa un punto del espacio de soluciones del problema.

Dado que un algoritmo genético trabaja sobre *cromosomas*, se debe definir una *función de codificación* sobre los puntos del espacio de soluciones, que mapea todo punto del espacio de soluciones en un *genotipo*. La *función inversa* de la codificación, denominada *decodificación* permite obtener el *fenotipo* asociado a un *cromosoma*.

Tomando en cuenta la observación anterior, los mecanismos de codificación de individuos solución resultan importantes para el proceso de búsqueda de los algoritmos genéticos. Habitualmente los algoritmos genéticos utilizan codificaciones binarias de largo fijo. Los individuos se codifican por un conjunto de cardinalidad conocida de valores binarios (ceros y unos) conocido como *string* de bits o *bitstring*. Cada *bitstring* representa a una solución potencial del problema de acuerdo al mecanismo de codificación predefinido, en general dependiente del problema. Otros esquemas de codificación han sido utilizados con menor frecuencia en los algoritmos genéticos. En particular las codificaciones basadas en números reales son útiles para representar soluciones cuando se resuelven problemas sobre espacios de cardinalidad no numerable, como en el caso de determinación de parámetros en problemas de control o entrenamiento de redes neuronales. Los esquemas basados en permutaciones de enteros son útiles para problemas de optimización combinatoria que involucran hallar ordenamientos óptimos, como los problemas de *scheduling* o el reconocido Traveling Salesman Problem ((Lawler and Col 1985)).

Codificaciones dependientes de los problemas se han propuesto con frecuencia, como un mecanismo que permite incorporar conocimiento específico en la resolución de problemas complejos.

La Figura 1.1 resume gráficamente la relación entre el espacio de soluciones de un problema, la población de cromosomas con la cual trabaja el algoritmo genético y las funciones de codificación y decodificación.

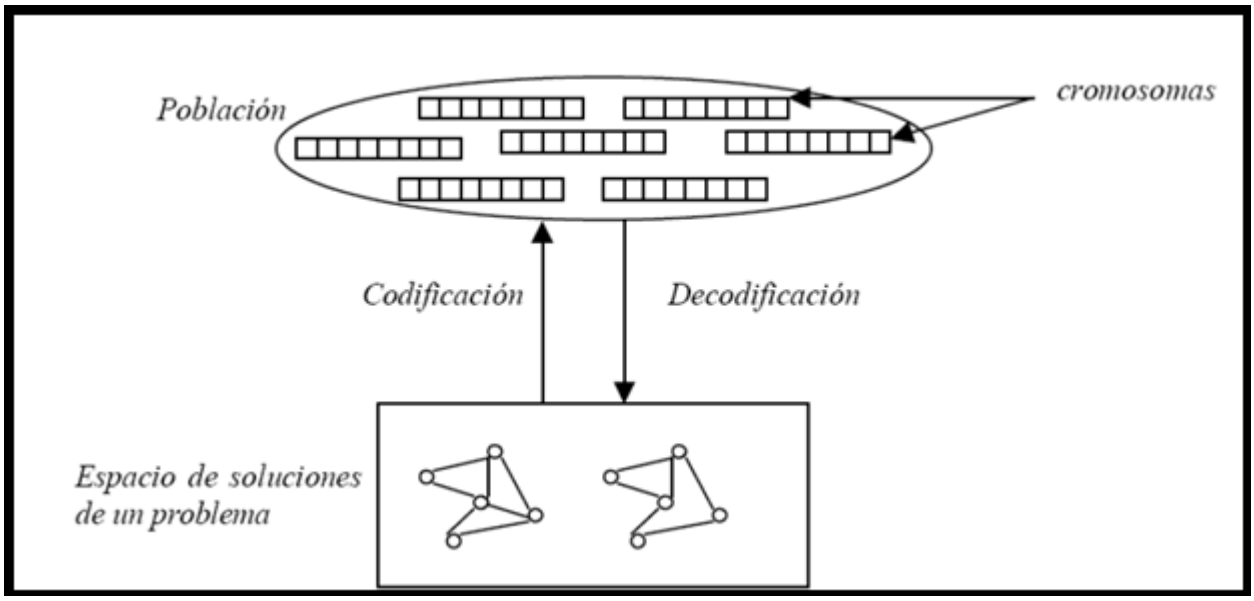


Figura 1.1: Codificación de soluciones en un algoritmo genético

1.1.2 Función de evaluación

Todo cromosoma tiene un valor asociado de fitness que evalúa aptitud del individuo para resolver el problema en cuestión. La función de fitness tiene el mismo tipo que la función objetivo del problema, lo cual implica que el cálculo del valor de fitness se realiza sobre el fenotipo correspondiente al cromosoma, como se presenta en la Figura1.2.

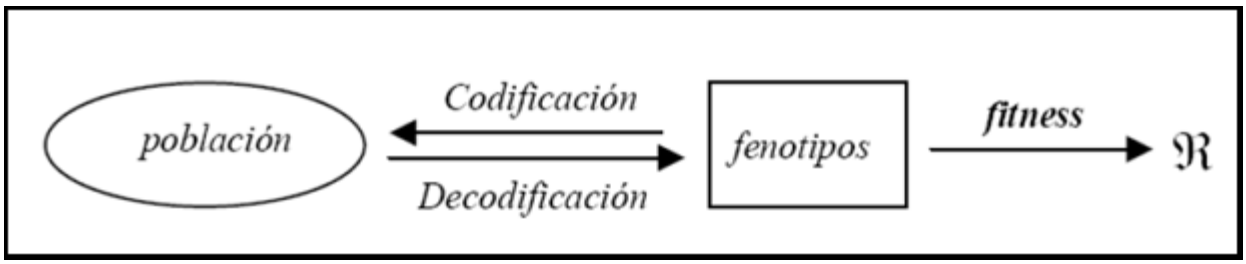


Figura 1.2: Codificación y función de fitness

La función de fitness tiene una influencia importante en el mecanismo del AG. Si bien actúa como una caja negra para el proceso evolutivo, la función de fitness guía el mecanismo de exploración, al actuar representando al entorno que evalúa la bondad de un individuo solución para la resolución del problema.

Varios puntos han sido identificados como importantes al momento de construir la función de fitness. De acuerdo a (Alba and Cotta 2003), es posible indicar que:

- La función de fitness debe contemplar el criterio del problema de optimización (minimización o maximización de un objetivo) y las restricciones presentes en el problema de optimización. En caso de surgir soluciones no factibles del problema, la función de fitness deberá asignarle valores adecuados que garanticen que tales individuos no se perpetúen durante el proceso evolutivo (valores muy pequeños en el caso de un problema de maximización y muy elevados en el caso de un problema de minimización). A tales efectos diversas transformaciones a aplicar sobre funciones de fitness han sido definidas para mapear problemas de minimización a maximización y aplicar penalizaciones a soluciones no factibles (GOLDBERG 1989).
- Deben contemplarse los casos en que el entorno presente problemas para la evaluación de la función de fitness, utilizando evaluaciones parciales cuando existen valores de fitness no definidos. Asimismo, debe considerarse el uso de funciones de fitness multivaluadas que asignen diferentes valores a un mismo individuo para simplificar la labor del operador de selección.
- El caso de funciones de fitness que varían dinámicamente durante la evolución del algoritmo genético debe tenerse en cuenta los mecanismos complejos como las representaciones múltiples, que son útiles para introducir memoria en la operativa del algoritmo genético.
- En caso de problemas con objetivos múltiples, todos ellos deben estar contemplados en la función de fitness. La utilización de algoritmos genéticos para la resolución de problemas multiobjetivo constituye en sí misma una subárea de investigación con complejidades inherentes.
- Para resolver problemas de dominancia de soluciones muy adaptadas en generaciones tempranas de la evolución, y para evitar el estancamiento en poblaciones similares al final de la evolución, deben considerarse mecanismos de *escalado* de los valores de fitness (MICHALEWICZ 1992).

En general, la función de fitness será compleja de evaluar, en particular demandará un esfuerzo computacional mucho mayor que el requerido para realizar los operadores evolutivos. Inclusive podría ocurrir que el proceso de evaluación sea tan complejo que solamente valores

aproximados pudieran obtenerse en tiempos razonables. Este aspecto será importante al momento de proponer técnicas de alto desempeño para mejorar la eficiencia de los algoritmos genéticos.

1.1.3 Operadores

La gran mayoría de las variantes de algoritmos genéticos utiliza como operadores a la selección, la recombinación y la mutación. El mecanismo de selección determina el modo de perpetuar *buenas* características, que se asumen son aquellas presentes en los individuos más adaptados. El mecanismo de *selección proporcional* o *selección por ruleta* elige aleatoriamente individuos utilizando una ruleta sesgada, en la cual la probabilidad de ser seleccionado es proporcional al fitness de cada individuo. Otros mecanismos de selección introducen diferentes grados de elitismo, conservando un cierto número prefijado de los mejores individuos a través de las generaciones. En el caso de la *selección por torneo* se escogen aleatoriamente un determinado número de individuos de la población, los cuales compiten entre ellos para determinar cuáles se seleccionarán para reproducirse, de acuerdo a sus valores de fitness. El mecanismo *de selección basado en el rango* introduce el mayor grado de elitismo posible, al mantener entre generaciones un porcentaje generalmente elevado, de los mejores individuos de la población.

Estas diferentes políticas de selección, conjuntamente con políticas similares utilizadas para determinar los individuos reemplazados por los descendientes generados posibilitan el diseño de diferentes modelos evolutivos para los algoritmos genéticos.

Los esquemas de codificación binaria de largo fijo tienen como ventaja principal que resulta sencillo definir operadores evolutivos simples sobre ellos. En la formulación clásica de un algoritmo genético, denominada por Goldberg como *Algoritmo Genético Simple* (GOLDBERG 1989), se propone como operador de recombinación *el cruzamiento de un punto*, que consiste en obtener dos descendientes a partir de dos individuos padres seleccionando un punto al azar, cortando los padres e intercambiando los trozos de cromosoma, tal como se presenta en la Figura1.3.

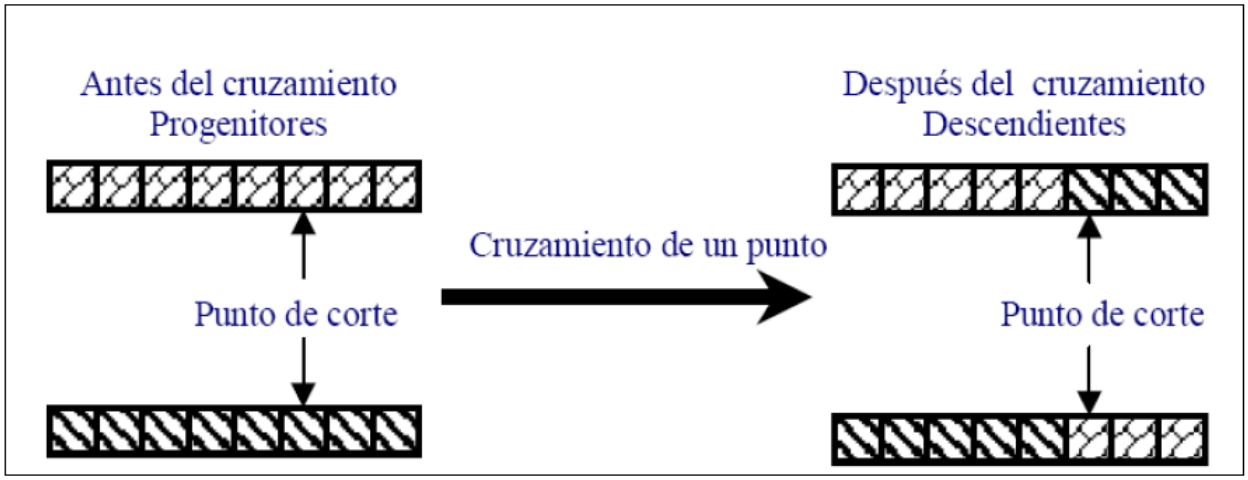


Figura 1.3: Esquema del cruzamiento de un punto

El operador tradicional de mutación introduce diversidad en el mecanismo evolutivo, simplemente modificando aleatoriamente uno de los valores binarios del cromosoma. Sobre un esquema de codificación binaria la modificación consiste en invertir el valor binario de un alelo, y por ello recibe el nombre de mutación de inversión del valor de un *bit (flip bit)*; su operativa se ejemplifica en la Figura 1.4.

Figura 145: Esquema de la mutación de inversión del valor de un alelo

Tanto la recombinación como la mutación son operadores probabilísticos, en el sentido en que se aplican o no, teniendo en cuenta una tasa de aplicación del operador. Generalmente la tasa de aplicación del operador de cruzamiento es elevada en un algoritmo genético simple (entre 0,5 y 0,9) mientras que la tasa de aplicación del operador de mutación es muy baja, del orden de 0,001 para cada bit en la representación.

Operadores evolutivos más complejos han sido propuestos como alternativas para modificar el comportamiento del mecanismo de exploración del espacio de soluciones. Es habitual encontrar operadores de cruzamiento multipunto en donde se utilizan dos o más puntos de corte (la operativa del cruzamiento de dos puntos se ejemplifica en la Figura 1.5) o uniformes en

donde para cada posición en el cromosoma se decide intercambiar material genético de acuerdo a una probabilidad prefijada (su operativa se ejemplifica en la Figura 1.6).

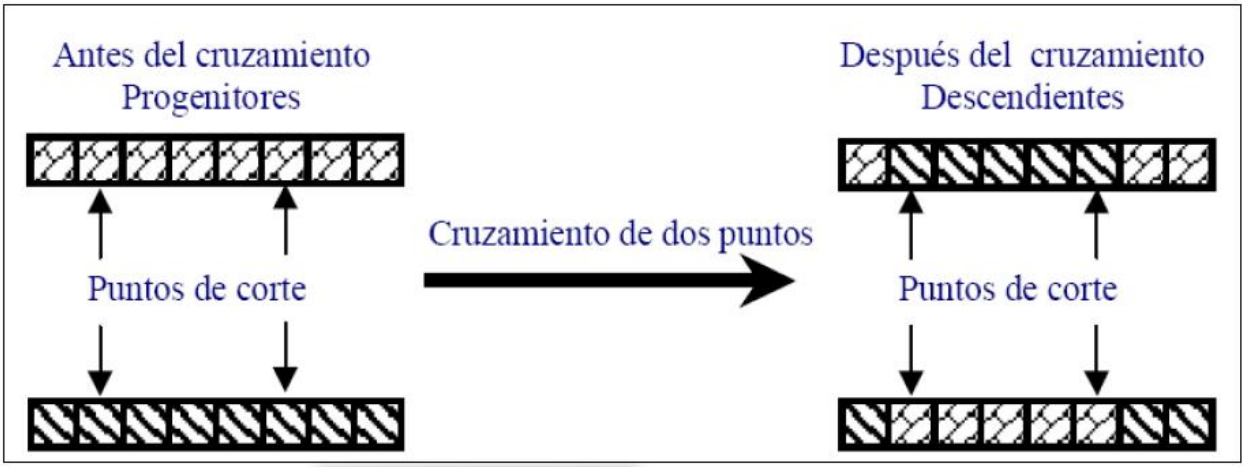


Figura 15: Esquema del cruzamiento de dos puntos

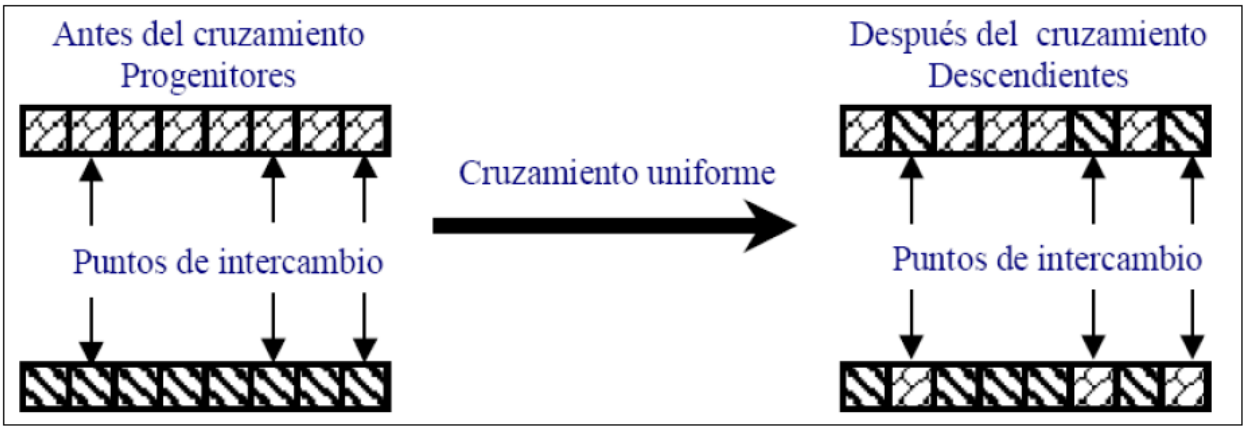


Figura 1.6: Esquema del cruzamiento uniforme

1.1.4 Modelos de evolución

En el modelo tradicional de algoritmos genéticos el paso básico de evolución consiste en el reemplazo de la totalidad de la población por sus descendientes en cada generación (*modelo generacional*). Otros modelos diferentes de evolución han sido propuestos por los investigadores en el área. Entre ellos, es posible destacar el modelo de *estado estacionario* (WHITLEY 1991). En este paradigma se crea un único nuevo individuo en cada generación, tratando de balancear el compromiso entre exploración (dada por el conjunto de individuos que forman la población) y explotación (realizada por el mecanismo de generación del único individuo creado en cada generación, generalmente utilizando técnicas de elitismo) para la resolución del problema.

Un modelo intermedio entre el tradicional y el de estado estacionario es el *modelo de gap generacional* (DE JONG and SARMA 1993), en el cual un porcentaje de la población (denominado *gap*) se genera en cada paso evolutivo.

Otros modelos de evolución más complejos han sido formulados para los algoritmos genéticos (modelos jerárquicos, no heterogéneos, híbridos, inspirados en los métodos de las estrategias de evolución). Estos modelos se diferencian de los tradicionales, incluyendo complejidades inherentes al mecanismo evolutivo que proponen.

1.1.5 Formalización del mecanismo de los algoritmos genéticos

La teoría tradicional de exploración del espacio de soluciones por parte del mecanismo evolutivo de los algoritmos genéticos fue formalizada por((Holland 1975)). Holland definió el concepto de *esquema*, para representar a conjuntos de individuos con características comunes de codificación y enunció el *teorema de los esquemas* para formalizar el muestreo de hiperplanos del espacio de soluciones. Como consecuencia de este teorema se propuso la denominada *hipótesis de los building blocks* que fundamenta el éxito del mecanismo de exploración de los algoritmos genéticos en el número exponencial de muestras que reciben aquellos esquemas cortos, con pocas posiciones definidas, pero que tienen un valor promedio de fitness superior al del resto de esquemas de la población. El mecanismo de recombinación, fundamental en el proceso evolutivo de un algoritmo genético, tenderá a crear individuos cada vez más aptos, perpetuando aquellas características adecuadas y combinándolas con otras de similar calidad (Holland 1975). Trabajos teóricos posteriores han complementado las estimaciones originales de Holland (WHITLEY 2002), e inclusive se han presentado formalizaciones matemáticas exactas del mecanismo, aunque también ofrecido puntos de vista alternativos y críticos a la *hipótesis de los building blocks*.

En (GOLDBERG, DEB et al. 1991) se demostró que el tiempo que un algoritmo genético emplea en converger hacia una solución única depende del tamaño de la población considerada. Utilizando los mecanismos de selección adecuados −aquellos que favorecen la reproducción de los mejores individuos−, el tiempo de convergencia del algoritmo genético es del orden de O(n log n), siendo n el tamaño de la población utilizada.

Adicionalmente a sus estudios sobre la formación de bloques básicos de construcción de soluciones, Holland se percató que el mecanismo evolutivo de los algoritmos genéticos tiene una

característica denominada paralelismo intrínseco, mediante la cual el ciclo evolutivo procesa un número mayor de esquemas que la cantidad de individuos presentes en la población. En efecto, trabajando sobre una población de n individuos, y por tanto requiriendo un esfuerzo computacional para realizar solamente n evaluaciones de la función de fitness, el algoritmo genético procesa útilmente O (n3) esquemas en cada generación. Por este motivo, al examinar un elevado número de secciones del espacio de soluciones, el mecanismo de los algoritmos genéticos logra compensar la disrupción de esquemas de mayor largo y de alto número de posiciones definidas cuando se aplican los operadores evolutivos de cruzamiento y mutación.

1.1.6 ¿Cuándo se pueden aplicar los Algoritmos Genéticos?

En primer lugar, en el problema a resolver la meta ha de poder ser observada en grados cualitativamente comparables. Por otra parte, las principales dificultades a la hora de implementar un algoritmo genético son:

- Definir una estructura de datos que pueda contener patrones que representen, la solución óptima buscada (desconocida) y todas las posibles alternativas de aproximaciones a la solución.
- Definir un tipo de patrón tal que si un patrón es seleccionado positivamente, que esto no sea debido a la interacción de los distintos segmentos del patrón, sino que existan segmentos que por sí solos provocan una selección positiva.
- Definir una función de evaluación que seleccione los mejores individuos.

Las condiciones que debe cumplir un problema para ser abordable por algoritmos evolutivos son:

- El espacio de búsqueda deber ser acotado.
- Debe existir un procedimiento relativamente rápido que asigne un grado de utilidad a cada solución propuesta, de forma que este grado de utilidad asignado corresponda, o bien directamente con la calidad de la solución en cuanto al problema a resolver, o bien con un valor de calidad relativo al resto de la población que permita obtener en el futuro mejores soluciones.
- Debe existir un método de codificación de soluciones que admita la posibilidad de que los cruzamientos combinen las características positivas de ambos progenitores. Este método debe permitir también aplicar algún mecanismo de mutación que sea capaz de

conseguir tantas soluciones muy dispares respecto de la solución sin mutar, como muy parecidas a ésta.

1.1.7 Comparación de los algoritmos genéticos con otras técnicas heurísticas

Si bien no existe una opinión unánime sobre la aplicabilidad de los algoritmos genéticos, la amplia gama de problemas resueltos exitosamente en diversas áreas han popularizado a esta técnica evolutiva como una de las más versátiles y robustas. Las opiniones de los investigadores en el área concuerdan en señalar a los algoritmos genéticos como altamente útiles para resolver problemas con espacio de soluciones de dimensión elevada o no muy bien comprendido de antemano, donde el diseño de operadores específicos de *hill-climbing* no sea sencillo, en problemas multimodales donde los métodos heurísticos tradicionales pueden quedar atrapados en óptimos locales o en casos donde no es necesario obtener una solución óptima al problema, sino que una buena solución aproximada sería suficiente.

Los algoritmos genéticos presentan ciertas características que los hacen ventajosos cuando se los compara con otras técnicas de resolución de problemas. La operativa del mecanismo evolutivo de los algoritmos genéticos es independiente de particularidades del dominio, permitiendo definir esquemas genéricos capaces de abordar diferentes clases de problemas. Esta característica, conjuntamente con el hecho de que el mecanismo evolutivo se aplique sobre representaciones, hace a los algoritmos genéticos aplicables a una amplia gama de problemas.

Además, los algoritmos genéticos no son sensibles a efectos de no linearidad de la función a optimizar o características del problema que afectan a algoritmos estándar de optimización que utilizan información adicional, como las técnicas de hill-climbing o algoritmos que asumen aspectos de linearidad, convexidad, diferenciabilidad u otras características sobre la función a optimizar. Estas características brindan a los algoritmos genéticos una significativa robustez de funcionamiento y de aplicabilidad.

Desde el punto de vista de la resolución del problema, el algoritmo genético funciona como una caja negra que solamente utiliza como dato de entrada la función de fitness definida para el problema para evaluar a los individuos en el ciclo evolutivo. No utiliza información adicional sobre las características del problema, aunque ciertas particularidades de la codificación pueden complementar la operativa simple de los operadores evolutivos y dirigir la búsqueda. La independencia del mecanismo de búsqueda evolutivo de las características del problema permite

a los algoritmos genéticos evitar, en ocasiones, el estancamiento en mínimos locales del problema en los cuales son proclives a caer ciertos algoritmos tradicionales basados en gradientes u otras búsquedas locales dirigidas.

Una descripción detallada del mecanismo operativo de los algoritmos genéticos, que complemente los breves detalles que se resumen en este capítulo, puede encontrarse en los textos de (GOLDBERG 1989) o (MITCHELL 1996).

1.1.8 Herramientas y bibliotecas evolutivas

La gran expansión que está sufriendo la Computación Evolutiva indica que las herramientas que se utilizan para el desarrollo deben ser cada día más flexibles y potentes, permitiendo a los investigadores construir cualquier tipo de estructura de datos que se desee implementar. Con esta flexibilidad, se facilita el hecho de que un investigador trabaje siempre con la misma herramienta aunque el problema que estén resolviendo en cada momento sea totalmente diferente al anterior, ahorrando tiempo de comprensión de nuevas herramientas (ARENAS, FOUCART et al.) y permitiendo la reutilización del trabajo ya terminado.

Aunque Java está considerado como un Lenguaje que cubre todos los ámbitos no existe ninguna biblioteca comparable en todos los puntos a Evolving Objects (MERELO, ARENAS et al. 2000) (MERELO, ARENAS et al. 2000) que tiene como principal norma de diseño la capacidad de poder hacer evolucionar cualquier tipo de estructura de datos (MERELO, KEIJZER et al.) o a GAlib (WALL 1995) ambas desarrolladas en su totalidad en C++.

Existen herramientas como pueden ser JGAP (ROTSTAN and MEFFERT 2008) (ROTSTAN and MEFFERT 2008) y JDEAL (COSTA, LOPES et al.), (COSTA, LOPES et al.), que incluye muchas de las características de EO, incluyendo distribución de código y muchos tipos diferentes de cromosomas y operadores. Sin embargo, (COSTA, LOPES et al.)no presenta la generalidad que EO posee, en el sentido de que sólo ofrece dos tipos de operadores: mutación y cruce y no parece estar diseñadas para ser totalmente ampliable puesto que su diseño está totalmente basado en clases no en interfaces. Esto limita al usuario la posibilidad de que el cromosoma o individuo que él necesita pueda ser, a la vez, de cromosoma y de operador, puesto que en Java no está permitido la herencia múltiple. Sería totalmente imposible que una misma clase heredara de algún tipo de cromosoma y algún tipo de operador a la vez. Pero no estaría de más decir que esta característica es simplemente una limitación, no un defecto.

Bastante recomendable en el caso que nos concierne, es utilizar una herramienta como ECJ. De este modo separaríamos claramente la parte del experimento de los detalles de implementación para la programación genética.

1.1.9 ECJ Características Generales

ECJ siglas en inglés para A Java-based Evolutionary Computation and Genetic Programming Research System (LUKE, PANAIT et al.) es una plataforma de desarrollo de algoritmos evolutivos desarrollada en el Laboratorio de Computación Evolutiva ECLab de la George Mason University (ECLAB 2010), casi todas sus clases y parámetros se determinan en tiempo de ejecución mediante archivos de parámetros jerárquicos.

Diseñada para ser flexible, hereda todas las ventajas de Java, como multiplataforma, multihilo, manejo de excepciones, garbage collection, incluye funciones de logging y checkpoint independientes para varias técnicas evolutivas. Brinda compatibilidad con Algoritmos Genéticos y Programación Genética de estado estable y/o generacional, con o sin elitismo para evolución de tipo: mu, lambda y mu+lambda sobre una arquitectura de reproducción (breeding) bastante flexible y modelos de islas asincrónicos sobre TCP/IP. Es capaz de manejar gran número de operadores de selección para múltiples subpoblaciones y especies, permitiendo intercambios entre subpoblaciones de individuos con genes de longitud fija o variable y soporta coevolución de población simple y múltiple, serialización de poblaciones en archivos persistentes y optimización multiobjetivo SPEA2, facilitando la posibilidad de extender la librería para otros métodos de optimización multiobjetivo.

1.2 Estrategias QSPR/QSAR

Lo modelos que relacionan la propiedad y la estructura cuantitativa (Karelson 2000) describen una relación matemática entre las características estructurales y una propiedad de una serie de compuestos químicos. El uso de tales relaciones matemáticas para predecir la propiedad objeto de interés en una sustancia química resulta seductor en lugar de mediciones experimentales laboriosas que conllevan un gran consumo de tiempo y recursos materiales muy costosos.

Los estudios QSPR constituyen un enfoque que permite entender como la variación estructural afecta la propiedad de un conjunto de compuestos. En estos estudios, los descriptores moleculares (X) se correlacionan con una variable respuesta (Y). Es decir, este análisis puede

definirse como una aplicación de métodos matemáticos y estadísticos al problema de encontrar una ecuación empírica de la forma $Y_i = f_i(X_1, X_2,\ldots X_n)$, donde Y_i son las propiedades de la molécula y $X_1, X_2\ldots X_n$, son propiedades estructurales experimentales o calculadas (DM) de los compuestos. En este sentido, cada compuesto puede representarse como un punto en un espacio multidimensional, en los cuales los descriptores $X_1, X_2\ldots X_n$ son coordenadas independientes del compuesto. El objetivo más usual de este análisis es predecir la propiedad estudiada a un objetivo (compuesto) no utilizado en la obtención del modelo.

La principal característica de estas estrategias es el enfoque multivariado al problema, la búsqueda de información relevante, la validación de los modelos para generar modelos con poder predictivo, comparación de los resultados obtenidos por diferentes métodos, y la definición y el uso de índices capaces de medirla calidad de la información extraída. Esta es la herramienta más usada en los estudios QSPR ya que brinda una sólida base para el análisis y la modelación de datos proporcionando una batería de diferentes métodos para este fin. Varias técnicas estadísticas fueron aplicadas para predecir propiedades, usando descriptores moleculares, pero las más utilizadas son el análisis de Cluster, la regresión lineal múltiple y las redes neuronales (CHEN 2008).

Con el objetivo de unificar estos criterios en todas las investigaciones del tema surgen los principios de la Organización para la Económica, Cooperación y el Desarrollo (OECD 2004), que por su gran interés fueron considerados durante el desarrollo del trabajo. Estos principios de validación de la OECD (OECD 2004 2007), se crean para facilitar las consideraciones de los modelos obtenidos y están asociados con la siguiente información:

1. Punto final definido (variable Y dependiente).
2. Algoritmo no ambiguo.
3. Dominio de aplicabilidad definido.
4. Mediciones apropiadas de bondad de ajuste, robustez y predictividad.
5. Interpretación mecanicista, si es posible.

1.2.1 Metodología general empleada en estudios QSPR/QSAR

Solo los modelos QSPR validados pueden ofrecer una interpretación significativa, especialmente en el contexto de diseñar o descubrir nuevos agentes químicos con propiedad

deseada. Los principios de la metodología QSAR/QSPR puede describirse mediante los siguientes pasos comunes (van de Waterbeemd 1995):

1. Formulación del problema, se determina el objeto de análisis y nivel de información requerido.
2. Parametrización cuantitativa de la estructura molecular de los compuestos químicos orgánicos.
3. Medición de la propiedad de interés (efectos biológicos, índices de retención, toxicidad).
4. Escoger el tipo de modelo QSAR/QSPR que se va a desarrollar.
5. Selección de los compuestos.
6. Análisis matemático de los datos y validación interna y externa de los modelos obtenidos.
7. Definición del dominio de aplicación del modelo obtenido.

1.2.2 Validación de modelos

Independientemente del tipo de análisis QSAR empleado el resultado final va a ser un modelo matemático. Para estar completamente seguros de que el modelo obtenido no resulta de un ordenamiento a azar de los datos, es común verificar la calidad del mismo según las normativas internacionales.

Otro problema es que el modelo no sea predictivo, es decir que la función propuesta no cumpla para productos no incluidos en la serie de exploración. Esta problemática suele solucionarse realizando una validación cruzada (*crossvalidation*) y el Split con un 66%.

Para que el estudio de QSAR se concluya con éxito debe obtenerse un modelo significativo y predictivo. Aun así, la validez del modelo estará siempre limitada al rango de parámetros explorados por la serie de exploración, fuera del cual nunca debe considerarse válido. De hecho, esta es la principal limitación a la hora de buscar el óptimo de actividad predicho por un modelo.

1.2.3 Herramientas y bibliotecas de química y biología

A medida que aumenta los problemas en la Quimica Computacional y en la Biologia Computacional, tener herramientas potentes es una de las necesidades de estas ramas de la ciencia.

Hay muchas herramientas computacionales como el (Tubert-Brohman) – (Perl Modules for Molecular Chemistry), (MayaChem 2010) es una colección en crescimiento de script de Perl, modulos, clases que son suportados dia a dia en lo que se necesita en la descuberta

computacional, (O'Boyle and Hutchison 2008) presenta un API común para herramientas de la química computacional, (ChemoJava 2009) es un proyecto basado en el la plataforma CDK, donde agregaron agregaron más funcionalidades de la química computacional pero sigue usando la misma API, CDL (CDL 2009) brindanos una genérica plataforma en C++ para escribir algoritmos para el calculo de descriptores moleculares, también oferta una eficiente estructura de búsqueda, algoritmos de huellas y pharmacophore, y otras más cosas para el cálculo de descriptores, el Chemistry Developtment Kit (STEINBECK, HAN et al. 2003) es una librería en Java para la química y biología computacional, actualmente desarrollada por mas de 50 programadores de todo el mundo y usada por mas de 10 universidades como los proyectos empresariales de todo el mundo.

Bastante recomendable en el caso que nos concierne, es utilizar una herramienta como CDK.

1.2.4 CDK Características Generales

La plataforma fue desarrollada por Christoph Steibeck, Egon Willighagen y Dan Gezelter de la Universidad de Notre Dame, Sur Bend, USA en 27-29 de Septiembre del 2000.

Diseñada en Java, hereda todas las facilidades de java, suporta estrutura eficientes de búsqueda, teoría de grafos y todas las herramientas para el manejo de la estrutura química y biológica.

1.3 Algoritmos genéticos aplicados a la Química Computacional

Adentrándonos en la química computacional, nos encontramos con un número considerable de aplicaciones, que en su mayoría presentan un espacio de búsqueda de tamaño exponencial directamente proporcional a las dimensiones del problema, por tanto estos problemas no pueden ser resueltos por métodos tradicionales de búsqueda exhaustiva. Paradójicamente los algoritmos genéticos (GOLDBERG 1989) presentan una adecuada efectividad en los espacios de búsqueda considerados “grandes”.

En los últimos años la aplicación de los AG en estos complejos problemas se ha convertido en un amplio tema de desarrollo y existe una creciente colección de publicaciones que así lo demuestran. Como podemos apreciar (CLARK and WESTHEAD 1996), con el diseño de moléculas auxiliado por computadoras (computer-aided molecule design), (WILLETT 1995) con el reconocimiento de moléculas, (PARRILL 1996) en el descubrimiento de fármacos, (PEDERSEN and MOULT 1996) con la predicción de las estructuras en proteínas, finalmente

podemos encontrar en (DEVILLERS 1996) un estudio general de las aplicaciones sobre el modelado de moléculas, entre otros.

1.4 Descriptores moleculares

1.4.1 Definición

Los índices o descriptores moleculares (DM) representan la vía por la cual la estructura química se transforma en números permitiendo el tratamiento matemático de la información química contenida en la molécula y pueden definirse como representaciones matemáticas de las moléculas que se obtienen al aplicar algoritmos específicos sobre una representación molecular definida o a partir de procedimientos experimentales específicos.

La utilidad de un DM debe analizarse con doble sentido: el número puede brindar una interpretación más profunda en términos estructurales de la propiedad molecular y/o es capaz de tomar parte en un modelo para la predicción de propiedades moleculares de interés (Todeschini and Consonni 2000). Incluso si la interpretación del DM es débil o carente, este podría estar estrechamente correlacionado con algunas propiedades moleculares permitiendo obtener modelos con alta capacidad predictiva. Por otro lado, DM con baja capacidad predictiva pueden ser mantenidos en el modelo cuando están correctamente fundamentados por la teoría y son interpretables debido a su capacidad para codificar información de la estructura química.

Aunque hasta el momento el número de índices moleculares reportados supera el millar, el desempeño de estos en la predicción de determinadas propiedades no siempre es totalmente satisfactorio. Por esa razón, la definición de DM se mantiene como un área de intensa actividad en el campo de la química computacional. La aplicación de conceptos de la Teoría de Grafos (Busacker and Saaty 1965) en el desarrollo de métodos teóricos para la representación de estructuras químicas ha tenido un enorme impacto, entre las aplicaciones más importantes de la TG a la química se encuentra la caracterización numérica de la estructura molecular a partir de invariantes grafo-teóricas, que se emplean como DM de compuestos químicos en estudios de estructura-propiedad (Diudea 2001).

1.4.2 Tipos

La naturaleza de los descriptores, depende de cual haya sido el proceder utilizado para la definición de los mismos, pudiendo tener en cuenta rasgos topológicos (Marrero-Ponce and

Torrens 2006), geométricos (Castillo-Garit, Marrero-Ponce et al. 2006), y electrónicos de las moléculas. Algunos de estos descriptores sin embargo, tienen "más información" de propiedades físico-químicas que de los rasgos estructurales de la molécula. Estos incluyen los basados en la determinación experimental de propiedades físico-químicas, tales como la mayoría de las constantes de los sustituyentes, hidrofobias, electrónicas y estéricas. (Kubinyi 1993) En contraste, los llamados índices topológicos (IT) tienen la información estructural contenida en una representación bidimensional de las moléculas, generalmente el grafo molecular con los átomos de hidrógenos suprimidos, sin considerar ningún rasgo físico-químico de las moléculas. La mayoría de estos índices pueden considerarse como descriptores estructurales explícitos. Otro grupo de descriptores, llamados químico-cuánticos describen rasgos electrónicos de las moléculas basados en el uso de la función de onda molecular. Los descriptores geométricos tienen información de los rasgos estructurales 3D de las moléculas en una vía explícita, tales como distancia y ángulos de enlaces o en una vía implícita, en forma de descriptores topográficos.

La próxima sección introduce un enfoque innovador basado en nociones de la TG y en conceptos básicos de álgebra matricial y vectorial aplicados en formas lineales, cuadráticas y bilineales.

1.4.3 Descriptores basados en formas algebraicas

1.4.3.1 Descriptores TOMOCOMD-CARDD "no estocástico"

En orden de obtener los descriptores ***TOMOCOMD-CARDD***, un vector molecular (**X**) es construido. Los componentes de este vector son valores numéricos de una propiedad que caracteriza cada tipo de átomo presente en la molécula.

Dado una molécula constituida por *n* átomos (vector de $\Re^n$), los k^{th} índices cuadráticos moleculares, $\boldsymbol{q_k}(x)$ son calculados como una forma cuadrática (q: $\Re^n \rightarrow \Re$) en las bases canónicas como se muestra en la Ec. 1.4.3.1.1:

$$q_k(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} {}^{k}a_{ij} X_i X_j \quad \textbf{(1.4.3.1.1)}$$

Donde ${}^{k}a_{ij} = {}^{k}a_{ji}$ (matriz cuadrada y simétrica), n es el número de átomos de la molécula y $X_1,\ldots,X_n$ son las coordenadas del vector molecular **X** en un sistema de vectores bases de $\Re^n$. Los valores de las coordenadas del vector dependen de la base escogida (Maltsev 1976; Noriega 1990; Axler 1996; Browder 1996). En la llamada base canónica (natural), e_j las coordenadas de cualquier vector coinciden con los componentes del vector (Maltsev 1976; Axler 1996; Browder 1996).Por esta razón, las coordenadas de los vectores pueden ser consideradas como pesos o etiquetas de átomos, permitiendo diferenciar entre heteroátomos.

Los coeficientes ${}^{k}a_{ij}$son los elementos a_{ij} de la k^{th} potencia de la matriz de adyacencia entre vértices ($\mathbf{M}^{k}$) del seudografo molecular (Bonchev and Rouvray 1991), la cual es utilizada como la matriz de la forma con respecto a la base canónica. Luego, **M** (Busacker and Saaty 1965) ≡ **M** = [a_{ij}], donde n es el número de vértices, y los elementos a_{ij}se definen como sigue:

$a_{ij} = P_{ij}$ si $i \neq j$ y $\exists\ e_k \in E\ /\ e_k \sim v_i, v_j$ **(1.4.3.1.2)**

$= L_{ii}$ si $i = j$

= 0 de otra forma

donde E(G) representa el conjunto de las aristas. P_{ij}es el número de aristas entre los vértices v_i y v_j. L_{ij}es el número de lazos en v_i.

Los elementos a_{ij} ($a_{ij} = P_{ij}$) de la matriz **M** representan los enlaces entre un átomo v_i y otro v_j. La matriz $\mathbf{M}^{k}$ provee el número de *caminos de longitud k* que une los vértices de v_i y v_j. Por esta razón, cada arista representa dos electrones del enlace covalente entre dos átomos v_i y v_j; y esto puede ser apreciado en las entradas a_{ij} y a_{ji}igual a 1, 2 y 3 de la matriz **M** (k = 1) cuando entre los átomos (vértices) v_i y v_j existe un simple, doble o triple enlace, respectivamente.

Las moléculas aromáticas como piridina, naftaleno, quinoleína, etc; donde existen más de una estructura canónica, los electrones de los orbitales PI (π) son representados como lazos sobre los átomos del anillo.

Los anillos aromáticos con una sola estructura canónica tales como el furano, tiofeno, pirrol, entre otros, son representados como *multigrafos*. Además, los $\boldsymbol{q_k}(x)$ pueden ser obtenidos mediante la expresión matricial representada en la Ec. 1.4.3.1.3:

$$q_k(x) = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}^k \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad \textbf{(1.4.3.1.3)}$$

o de forma más sencilla,

$\boldsymbol{q_k}(x) = [X]^t \mathbf{M}^{\boldsymbol{k}}[X]$ **(1.4.3.1.4)**

donde [X] es el vector columna (una matriz de *nx*1) de coordenadas de **X** en la base canónica de $\Re^n$, $[X]^t$ es la matriz transpuesta de [X] (una matriz de 1*xn*) y $\mathbf{M}^{\boldsymbol{k}}$ la k^{th} potencia de la matriz **M** de el seudografo molecular G (matriz de la forma cuadrática).

Uno de los criterios importantes de la lista de propiedades deseables para un nuevo IT es la posibilidad de definir localmente los descriptores (Randic 1991). Este atributo se refiere al hecho de que el índice no sea obtenido solamente de forma global para una estructura, sino que también puedan ser definidos sobre determinados fragmentos de la propia estructura.

Es por ello que se ha propuesto una definición local (para átomos o tipos de átomos) de los índices cuadráticos moleculares. Estos descriptores moleculares han sido denominamos como *índices cuadráticoslocales* de la matriz de adyacencia entre vértices de un seudografo molecular, $\boldsymbol{q_{kL}}(x)$ en donde se ha mantenido la generalización a "análogos superiores" (como una secuencia de números) (Randic 1991). La definición de estos descriptores, invariantes grafo-teóricas para un fragmento F_R dado, dentro de un seudografo específico (G) es la siguiente:

$$q_{kL}(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} {}^{k}a_{ijL} X_i X_j \quad \textbf{(1.4.3.1.5)}$$

Donde *m* es el número de átomos del fragmento de interés y $^{\boldsymbol{k}}\boldsymbol{a_{ijL}}$ es el elemento de la fila "*i*" y columna "*j*" de la matriz $\mathbf{M^k_L} = \mathbf{M^k}(G, F_R)$ [$\boldsymbol{q_{kL}}(x) = \boldsymbol{q_k}(x, F_R)$]. Esta matriz se extrae de la matriz k^{th} potencia de **M** y contiene la información referida a los vértices del fragmento $F_R$de interés y también de su entorno molecular.

La matriz $\mathbf{M^k_L} = [{}^{k}a_{ijL}]$ y los elementos ${}^{k}a_{ijL}$ son definidos como se muestra a continuación:

${}^{k}a_{ijL} = {}^{k}a_{ij}$ si tanto v_i como $v_j$son vértices contenidos en el fragmento de interés. **(1.4.3.1.6)**

$= {}^{1}/_{2}\,{}^{k}a_{ij}$si v_i o v_j están contenidos en el fragmento de interés pero no ambos

= 0 de otra forma

Siendo ${}^{k}a_{ij}$ los elementos de la k^{th} potencia de **M**. Estos análogos locales también pueden ser expresados de forma matricial mediante la siguiente expresión:

$\boldsymbol{q_{kL}}(x) = [X]^t \mathbf{M}^{k}_{\mathbf{L}} [X]$ **(1.4.3.1.7)**

Nótese que si una molécula es particionada en Z fragmentos moleculares, la matriz $\mathbf{M}^k$ puede ser particionada en Z matrices locales $\mathbf{M}^{\mathbf{k}}_{\mathbf{L}}$, L = 1,...Z. La matriz k^{th} potencia de **M** es exactamente la suma de las k^{th} potencia de las Z matrices locales,

$$\mathbf{M}^{\boldsymbol{k}} = \sum_{L=1}^{Z} M_{L}^{k} \quad \textbf{(1.4.3.1.8)}$$

o lo que es lo mismo $\mathbf{M}^{\mathbf{k}} = [{}^{k}a_{ij}]$ donde,

$$^{\boldsymbol{k}}\boldsymbol{a_{ij}} = \sum_{L=1}^{Z} {}^{k}a_{ijL} \quad \textbf{(1.4.3.1.9)}$$

y los índices cuadráticos totales son iguales a la suma de los índices cuadráticos locales de los Z fragmentos,

$$\boldsymbol{q_k}(x) = \sum_{L=1}^{Z} q_{kL}(x)$$

(1.4.3.1.10)

Cada orden de los índices cuadráticos locales tiene un significado particular, especialmente para los primeros valores de *k*, contienen información sobre la estructura del fragmento F_R en sí, para valores mayores, contiene información sobre el entorno del fragmento F_R considerado dentro del seudografo molecular (G).

Los k^{th} índices lineales locales (atómicos), $\boldsymbol{f_k}(x)$ son calculados como una aplicación lineal sobre $\Re^n$ [$\boldsymbol{f_k}(x)$: $\Re^n \rightarrow \Re^n$; entonces$\boldsymbol{f_k}(x)$: Endomorfismo sobre $\Re^n$] en las bases canónicas como se muestra en la ecuación ,

$$f_k(x_i) = \sum_{j=1}^{n} {}^{k}a_{ij} X_j \quad \textbf{(1.4.3.1.11)}$$

donde, ${}^{k}a_{ij} = {}^{k}a_{ji}$, *n* es el número de átomos de la moléculas y X_j son las coordenadas del vector molecular (**X**) en el sistema de bases canónicas de $\Re^n$. Los coeficientes ${}^{k}a_{ij}$son los elementos a_{ij} de la k^{th} potencia de la matriz de adyacencia entre vértices ($\mathbf{M}^{\boldsymbol{k}}$) del seudografo molecular (G). O sea, que **M**(G) denota la matriz de $\boldsymbol{f_k}(x)$ con respecto a la bases canónicas.

Nótese, que los índice lineales atómicos son definidos como una transformación lineal $\boldsymbol{f_k}(x)$ sobre un espacio vectorial molecular, $\Re^n$. Esta aplicación, es una correspondencia que asigna a cualquier vector **X** en $\Re^n$ un vector $\boldsymbol{f}(x)$ de forma tal que:

$$f(\lambda_1 X_1 + \lambda_2 X_2) = \lambda_1 f(X_1) + \lambda_2 f(X_2)$$

(1.4.3.1.12)

para todo λ_1,λ_2 número reales y cualquier vector X_1, X_2 en $\Re^n$. En otras palabras, $\boldsymbol{f_k}(x)$ es una aplicación lineal dado que la imagen de la combinación lineal de dos vectores X_1 y X_2, $\lambda_1 X_1 + \lambda_2 X_2$; es igual a la combinación lineal de las imágenes $f(X_1)$ y $f(X_2)$, $\lambda_1 f(X_1) + \lambda_2 f(X_2)$. Esta condición se denomina *condición de linealidad* (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996). La ecuación de definición (1.4.3.1.11) para $\boldsymbol{f_k}(x)$ puede ser escrita como una simple ecuación matricial:

$$f_k(x_i) = \begin{bmatrix} X_1' \\ \vdots \\ X_n' \end{bmatrix}^k = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}^k \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \textbf{(1.4.3.1.13)}$$

o de forma más sencilla,

$\boldsymbol{f_k}(x) = [X']^k = \mathbf{M}^k[X]$ **(1.4.3.1.14)**

Donde [X] es el vector columna (una matriz de *nx*1) de coordenadas de X en la base canónica de $\Re^n$, $[X]^t$ es la matriz transpuesta de [X] (una matriz de 1*xn*) y M^k la k^{th}potencia de la matriz M del seudografo molecular G (Matriz de la aplicación lineal).

Los índices lineales totales constituyen funciones lineales (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996). (Algunos matemáticos usan el término *formas lineales*) sobre $\Re^n$. Es decir, los k^{th} índices lineales totales constituyen aplicaciones lineales de $\Re^n$ a escalares $\Re$ [$f_k(x)$: $\Re^n \rightarrow \Re$]. La definición matemática de este descriptor molecular es la siguiente:

$$f_k(x) = \sum_{i=1}^{n} f_k(x_i) \textbf{ (1.4.3.1.16)}$$

Donde *n* es el número de átomos y $\boldsymbol{f_k}(x)$ son los índices lineales atómicos obtenidos por la Ec. 3.11. Entonces, una forma lineal $\boldsymbol{f_k}(x)$ puede ser escrita en forma matricial,

$\boldsymbol{f_k}(x) = [u]^t [X']^k$ **(1.4.3.1.17)**

o

$\boldsymbol{f_k}(x) = [u]^t \mathbf{M}^k[X]$ **(1.4.3.1.18)**

Para todo vector molecular $\mathbf{X} \in \Re^n$. $[u]^t$ es un vector fila (matriz fila) unitario de dimensión *n*.

Como puede observarse, los k^{th} índices lineales totales son calculados sumando todos los índices locales (átomos) de todos los átomos en la molécula.

Además, si una molécula es particionada en Z fragmentos moleculares, los índices lineales totales pueden ser particionados en Z índices lineales locales $f_{kL}(x)$, L = 1,…, Z. Es decir, los índices lineales totales de orden *k* pueden ser expresados como la suma de los índices locales de los Z fragmentos moleculares:

$$f_k(x) = \sum_{L=1}^{Z} f_{kL}(x) \quad \textbf{(1.4.3.1.19)}$$

Cada tipo de índice lineal local puede ser clasificado según el tipo de átomo que compone el fragmento. Así por ejemplo, se pueden calcular sobre heteroátomos, H-unido a heteroátomos (O, N y S), halógenos, cadenas alifáticas o aromáticas, entre otros.

Finalmente, los índices bilineales han sido definidos (Marrero-Ponce and Romero 2002) en analogía a las formas bilineales, los cuales son funciones más generales que los cuadráticos. Es decir, los índices cuadráticos constituyen un caso específico de los índices bilineales, o sea funciones asociadas con formas bilineales simétricas. Estos descriptores son definidos de forma similar a los índices cuadráticos (Ec. 1.4.3.1.1) según la ecuación 1.4.3.1.20.

$$b_k(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} {}^{k}a_{ij} X_i Y_j \quad \textbf{(1.4.3.1.20)}$$

Donde, ${}^{k}a_{ij} = {}^{k}a_{ji}$, *n* es el número de átomos de la moléculas. Los coeficientes ${}^{k}a_{ij}$son los elementos a_{ij} de la k^{th} potencia de la matriz de adyacencia entre vértices ($\mathbf{M}^k$) del seudografo molecular (G). X_i y Y_j son las coordenadas de los vectores moleculares, **X** y **Y** en el sistema de bases canónicas de $\Re^n$. Ambos vectores, **X** y **Y** tienen dimensión *n* (número de átomos) y pertenecen al mismo espacio vectorial $\Re^n$, su única diferencia entre ellos es el valor de sus elementos. En este sentido, se pueden calcular estos índices utilizando combinación de propiedades atómicas que caractericen diversos aspectos de los átomos de la molécula. Estos índices pueden ser obtenidos mediante la expresión matricial representada en la Ec. 1.4.3.1.21:

$$b_k(x) = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}^k \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \quad \textbf{(1.4.3.1.21)}$$

O de forma más sencilla,

$$\boldsymbol{b_k}(x) = [X]^t \mathbf{M}^k [Y] \quad \textbf{(1.4.3.1.22)}$$

Donde [Y] es el vector columna (una matriz de *nx*1) de coordenadas de **Y** en la base canónica de $\Re^n$, $[X]^t$ es la matriz transpuesta de [X], la cual es una matriz de 1*xn* v de coordenadas de **X** en la base canónica de $\Re^n$. $\mathbf{M}^k$ es la k^{th} potencia de la matriz **M** de el seudografo molecular G (matriz de la forma bilineal).

1.4.3.2 Descriptores TOMOCOMD-CARDD "estocásticos".

Los descriptores TOMOCOMD-CARDD pueden ser calculados en términos de probabilidades si utilizamos como matriz de las aplicaciones las matrices estocásticas de cada una de la potencias de la matriz de adyacencia entre átomos del seudografo molecular. Estos descriptores se han denominados como índices cuadráticos, lineales y bilineales estocásticos y presentan las mismas propiedades descritas para sus homólogos no estocásticos. Es decir, los *k*-ésimos descriptores moleculares estocásticos totales y locales se calculan según la mismas invariante definidas anteriormente [ver ecuaciones 1.4.3.1.1 (1.4.3.1.3), 1.4.3.1.11 (1.4.3.1.13) y 1.4.3.1.20 (1.4.3.1.21)], pero usando la matriz estocástica de adyacencia entre átomos del pseudografo molecular, $\mathbf{S}^k$(G), como matriz de las formas.

Como hemos planteado anteriormente, la matriz **M**(G) es una matriz de 'posicionamiento' de electrones. O sea, que **M**(G) es una matriz tipo-Hückel "extendido" en donde se consideran tanto los electrones sigma como pi (solo los electrones de la capa de valencia) que se comparten para formar enlaces covalentes. Una forma similar de expresar el compartimiento de electrones y más relacionada con la densidad electrónica es describir la probabilidad con que los átomos comparten sus electrones. Las matrices estocásticas $\mathbf{S}^k$(G) puede ser obtenidas dividiendo cada elemento ${}^k a_{ij}$ de $\mathbf{M}^k$(G) (electrones que comparte el átomo '*i*' con el átomo '*j*') entre el número de electrones que el átomo '*i*' comparte con todos los demás átomos en la molécula a *k*-distancias, incluido el mismo. Los elementos ${}^k\mathbf{s}_{ij}$ se definen como se muestra en la ecuación 1.4.3.2.1:

$$ {}^k s_{ij} = \frac{{}^k a_{ij}}{{}^k SUM_i} = \frac{{}^k a_{ij}}{{}^k \delta_i} \quad \textbf{(1.4.3.2.1)} $$

Donde ${}^k a_{ij}$ son los elementos de la *k*-ésima potencia de **M**, y ${}^k\mathrm{SUM}_i$ es la suma de la fila *i-ésima* de $\mathbf{M}^k$ o grado del vértice de orden *k* del átomo *i*, ${}^k\delta_i$. Esta transformación normaliza cada fila de la matriz original [las matrices con estas propiedades y con elementos no negativos se denominan estocásticas (Edwards and Penney 1988)] y por tanto, sus *k*-ésimos elementos

constituyen las probabilidades de transición con las cuales un electrón se mueve de un átomo *i* a otro *j* en un período de tiempo discreto t_k

1.4.3.2.1 Índices Cuadráticos Moleculares

Con el propósito de obtener los descriptores ***TOMOCOMD-CARDD***, un vector molecular (**X**) es construido. Los componentes de este vector son valores numéricos de una propiedad que caracterizan cada tipo de átomo presente en la molécula (tales como la electronegatividad, densidad, radio atómico, etc) (Marrero-Ponce and Romero 2002; Marrero-Ponce 2003; Marrero-Ponce 2004).

Dado una molécula constituida por *n* átomos (vector de $\Re^n$), los k^{th} índices cuadráticos moleculares, $\boldsymbol{q_k}(x)$ son calculados como una forma cuadrática (q: $\Re^n \rightarrow \Re$) en las bases canónicas como se muestra en la Ec. 3.1 (Marrero-Ponce and Romero 2002; Marrero-Ponce, Cabrera et al. 2003; Marrero-Ponce 2004; Marrero-Ponce 2004),

$$q_k(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} {}^k a_{ij} X_i X_j \quad \textbf{(1.4.3.2.1.1)}$$

Donde ${}^k a_{ij} = {}^k a_{ji}$ (Matriz cuadrada y simétrica),*n* es el número de átomos de la molécula y $X_1, \ldots, X_n$ son las coordenadas del vector molecular **X** en un sistema de vectores bases de $\Re^n$. Los valores de las coordenadas del vector dependen de la base escogida (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996). En la llamada base canónica (natural), e_j las coordenadas de cualquier vector coinciden con los componentes del vector (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996).Por esta razón, las coordenadas de los vectores pueden ser consideradas como pesos o etiquetas de átomos, permitiendo diferenciar entre heteroátomos.

Los coeficientes ${}^k a_{ij}$son los elementos a_{ij} de la k^{th} potencia de la matriz de adyacencia entre vértices ($\mathbf{M}^k$) del seudografo molecular (G), la cual es utilizada como la matriz de la forma con respecto a la base canónica. Luego, **M** (G) ≡ **M** = [a_{ij}], donde *n* es el número de vértices, y los elementos a_{ij}se definen como sigue (Marrero-Ponce and Romero 2002; Marrero-Ponce 2003; Marrero-Ponce 2004):

a_{ij} = P_{ij} si $i \neq j$ y $\exists$ $e_k \in E$ / $e_k \sim v_i, v_j$ **(1.4.3.2.1.2)**

= L_{ii} si $i = j$

= 0 de otra forma

donde E(G) representa el conjunto de las aristas. P_{ij}es el número de aristas entre los vértices v_i y v_j. L_{ij}es el número de lazos en v_i.

Los elementos a_{ij} ($a_{ij} = P_{ij}$) de la matriz **M** representan los enlaces entre un átomo v_i y otro v_j. La matriz $\mathbf{M}^k$ provee el número de *caminos de longitud k* que une los vértices de v_i y v_j. Por esta razón, cada arista representa dos electrones del enlace covalente entre dos átomos v_i y v_j; y esto puede ser apreciado en las entradas a_{ij} y a_{ji}igual a 1, 2 y 3 de la matriz **M** ($k = 1$) cuando entre los átomos (vértices) v_i y v_j existe un simple, doble o triple enlace, respectivamente.

Las moléculas aromáticas como piridina, naftaleno, quinoleína, etc; donde existen más de una estructura canónica, los electrones de los orbitales PI (π) son representados como lazos sobre los átomos del anillo.

Los anillos aromáticos con una sola estructura canónica tales como el furano, tiofeno, pirrol, entre otros, son representados como *multigrafos*. Además, los $\boldsymbol{q_k}(x)$ pueden ser obtenidos mediante la expresión matricial representada en la Ec. 1.4.3.2.1.3 (Marrero-Ponce and Romero 2002; Marrero-Ponce 2003; Marrero-Ponce 2004):

$$q_k(x) = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}^k \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad \textbf{(1.4.3.2.1.3)}$$

O de forma más sencilla,

$$\boldsymbol{q_k}(x) = [X]^t \mathbf{M}^{\boldsymbol{k}}[X] \quad \textbf{(1.4.3.2.1.4)}$$

Donde [X] es el vector columna (una matriz de $nx1$) de coordenadas de **X** en la base canónica de $\Re^n$, $[X]^t$ es la matriz transpuesta de [X] (una matriz de $1xn$) y $\mathbf{M}^{\boldsymbol{k}}$ la k^{th} potencia de la matriz **M** de el seudografo molecular G (Matriz de la forma cuadrática).

Uno de los criterios importantes de la lista de propiedades deseables para un nuevo IT es la posibilidad de definir localmente los descriptores (Randic 1991). Este atributo se refiere al hecho de que el índice no sea obtenido solamente de forma global para una estructura, sino que también puedan ser definidos sobre determinados fragmentos de la propia estructura.

Es por ello que se ha propuesto una definición local (para átomos o tipos de átomos) de los índices cuadráticos moleculares. Estos descriptores moleculares han sido denominamos como *índices cuadráticoslocales* de la matriz de adyacencia entre vértices de un seudografo molecular,

$\boldsymbol{q_{kL}}(x)$ en donde se ha mantenido la generalización a "análogos superiores" (como una secuencia de números) (Randic 1991). La definición de estos descriptores, invariantes grafo-teóricas para un fragmento F_R dado, dentro de un seudografo específico (G) es la siguiente (Marrero-Ponce and Romero 2002; Marrero-Ponce 2003; Marrero-Ponce 2004):

$$q_{kL}(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} {}^{k}a_{ijL} X_i X_j \quad \textbf{(1.4.3.2.1.5)}$$

Donde *m* es el número de átomos del fragmento de interés y $\boldsymbol{{}^{k}a_{ijL}}$ es el elemento de la fila "*i*" y columna "*j*" de la matriz $\mathbf{M^k_L} = \mathbf{M^k}(G, F_R)$ [$\boldsymbol{q_{kL}}(x) = \boldsymbol{q_k}(x, F_R)$]. Esta matriz se extrae de la matriz k^{th} potencia de **M** y contiene la información referida a los vértices del fragmento $F_R$de interés y también de su entorno molecular.

La matriz $\mathbf{M^k_L} = [{}^{k}a_{ijL}]$ y los elementos ${}^{k}a_{ijL}$ son definidos como se muestra a continuación:

${}^{k}a_{ijL} = {}^{k}a_{ij}$ si tanto v_i como $v_j$son vértices contenidos en el fragmento de interés. **(1.4.3.2.1.6)**

$= {}^{1}/_{2}\,{}^{k}a_{ij}$si v_i o v_j están contenidos en el fragmento de interés pero no ambos

= 0 de otra forma

siendo${}^{k}a_{ij}$los elementos de la k^{th} potencia de **M**. Estos análogos locales también pueden ser expresados de forma matricial mediante la siguiente expresión:

$\boldsymbol{q_{kL}}(x) = [X]^t \mathbf{M}^k_{\mathbf{L}} [X]$ **(1.4.3.2.1.7)**

Nótese que si una molécula es particionada en Z fragmentos moleculares, la matriz $\mathbf{M}^k$ puede ser particionada en Z matrices locales $\mathbf{M^k_L}$, L = 1,...Z. La matriz k^{th} potencia de **M** es exactamente la suma de las k^{th} potencia de las Z matrices locales,

$$\mathbf{M}^k = \sum_{L=1}^{Z} M_L^{\,k} \quad \textbf{(1.4.3.2.1.8)}$$

o lo que es lo mismo $\mathbf{M^k} = [{}^{k}a_{ij}]$ donde,

$$\boldsymbol{{}^{k}a_{ij}} = \sum_{L=1}^{Z} {}^{k}a_{ijL} \quad \textbf{(1.4.3.2.1.9)}$$

y los índices cuadráticos totales son iguales a la suma de los índices cuadráticos locales de los Z fragmentos,

$$\boldsymbol{q_k}(x) = \sum_{L=1}^{Z} q_{kL}(x) \quad \textbf{(1.4.3.2.1.10)}$$

Cada orden de los índices cuadráticos locales tiene un significado particular, especialmente para los primeros valores de *k*, contienen información sobre la estructura del fragmento F_R en sí, para valores mayores, contiene información sobre el entorno del fragmento F_R considerado dentro del seudografo molecular (G).

1.4.3.2.2 Índices lineales Moleculares

Dado una molécula constituida por *n* átomos (vector de $\Re^n$), los k^{th} índices lineales locales (atómicos), $\boldsymbol{f_k}(x)$ son calculados como una aplicación lineal sobre $\Re^n$ [$\boldsymbol{f_k}(x)$: $\Re^n \rightarrow \Re^n$; entonces$\boldsymbol{f_k}(x)$: Endomorfismo sobre $\Re^n$] en las bases canónicas como se muestra en la ecuación ,

$$f_k(x_i) = \sum_{j=1}^{n} {}^k a_{ij} X_j \quad \textbf{(1.4.3.2.2.1)}$$

donde, ${}^k a_{ij} = {}^k a_{ji}$, *n* es el numero de átomos de la moléculas y X_j son las coordenadas del vector molecular (**X**) en el sistema de bases canónicas de $\Re^n$. Los coeficientes ${}^k a_{ij}$son los elementos a_{ij} de la k^{th} potencia de la matriz de adyacencia entre vértices ($\mathbf{M}^k$) del seudografo molecular (G). O sea, que **M**(G) denota la matriz de $\boldsymbol{f_k}(x)$ con respecto a la bases canónicas.

Nótese, que los índice lineales atómicos son definidos como una transformación lineal $\boldsymbol{f_k}(x)$ sobre un espacio vectorial molecular, $\Re^n$. Esta aplicación, es una correspondencia que asigna a cualquier vector **X** en $\Re^n$ un vector ***f***(*x*) de forma tal que:

$f(\lambda_1 X_1 + \lambda_2 X_2) = \lambda_1 f(X_1) + \lambda_2 f(X_2)$ **(1.4.3.2.2.2)**

para todo λ_1,λ_2 número reales y cualquier vector X_1, X_2 en $\Re^n$. En otras palabras, $\boldsymbol{f_k}(x)$ es una aplicación lineal dado que la imagen de la combinación lineal de dos vectores X_1 y X_2, $\lambda_1 X_1 + \lambda_2 X_2$; es igual a la combinación lineal de las imágenes $f(X_1)$ y $f(X_2)$, $\lambda_1 f(X_1) + \lambda_2 f(X_2)$. Esta condición se denomina *condición de linealidad* (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996). La ecuación de definición (1.4.3.2.2.1) para $\boldsymbol{f_k}(x)$ puede ser escrita como una simple ecuación matricial:

$$f_k(x_i) = \begin{bmatrix} X'_1 \\ \vdots \\ X'_n \end{bmatrix}^k = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}^k \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad \textbf{(1.4.3.2.2.3)}$$

o de forma más sencilla,

$\boldsymbol{f_k}(x) = [X']^k = \mathbf{M}^{\boldsymbol{k}}[X]$ **(1.4.3.2.2.4)**

donde [X] es el vector columna (una matriz de *nx*1) de coordenadas de X en la base canónica de $\Re^n$, $[X]^t$ es la matriz transpuesta de [X] (una matriz de 1*xn*) y M^k la k^{th}potencia de la matriz M del seudografo molecular G (Matriz de la aplicación lineal).

Los índices lineales totales constituyen funcionales lineales (Maltsev 1976; Ross and Wright 1990; Axler 1996; Browder 1996). (Algunos matemáticos usan el termino *formas lineales*) sobre $\Re^n$. Es decir, los k^{th} índices lineales totales constituyen aplicaciones lineales de $\Re^n$ a escalares $\Re$ [$f_k(x)$: $\Re^n \rightarrow \Re$]. La definición matemática de este descriptor molecular es la siguiente:

$$f_k(x) = \sum_{i=1}^{n} f_k(x_i) \quad \textbf{(1.4.3.2.2.5)}$$

Donde *n* es el número de átomos y $\boldsymbol{f_k}(x)$ son los índices lineales atómicos obtenidos por la Ec. 3.11. Entonces, una forma lineal $\boldsymbol{f_k}(x)$ puede ser escrita en forma matricial,

$\boldsymbol{f_k}(x) = [u]^t [X']^k$ **(1.4.3.2.2.6)**

o

$\boldsymbol{f_k}(x) = [u]^t \mathbf{M}^{\boldsymbol{k}}[X]$ **(1.4.3.2.2.7)**

para todo vector molecular $\mathbf{X} \in \Re^n$. $[u]^t$ es un vector fila (matriz fila) unitario de dimensión *n*.

Como puede observarse, los k^{th} índices lineales totales son calculados sumando todos los índices locales (átomos) de todos los átomos en la molécula.

Además, si una molécula es particionada en Z fragmentos moleculares, los índices lineales totales pueden ser particionado en Z índices lineales locales $\boldsymbol{f_{kL}}(x)$, L = 1, …, Z. Es decir, los índices lineales totales de orden *k* pueden ser expresados como la suma de los índices locales de los Z fragmentos moleculares:

$$f_k(x) = \sum_{L=1}^{Z} f_{kL}(x) \quad \textbf{(1.4.3.2.2.8)}$$

Cada tipo de índice lineal local puede ser clasificado según el tipo de átomo que compone el fragmento. Así por ejemplo, se pueden calcular sobre heteroatomos, H-unido a heteroatomos (O, N y S), halógenos, cadenas alifáticas o aromáticas, entre otros.

1.4.3.2.3 Índices Cuadráticos y Lineales Estocásticos

Los k^{th} índices cuadráticos y lineales moleculares pueden ser calculados en términos de probabilidades si utilizamos como matriz de las aplicaciones las matrices estocásticas de cada una de la potencias de la matriz de adyacencia entre átomos del seudografo molecular. Como hemos planteado anteriormente, la matriz **M**(G) es una matriz de 'posicionamiento' de electrones. O sea, que **M**(G) es una matriz tipo-Hückel en donde se consideran tanto los electrones sigma como pi (solo los electrones de la capa de valencia) que se comparten para formar enlaces covalentes. Una forma similar de expresar el compartimiento de electrones y más relacionada con la densidad electrónica es describir la probabilidad con que los átomos comparten sus electrones. Las matrices estocásticas $\mathbf{S}^k$(G) puede ser obtenidas dividiendo cada elemento ${}^k a_{ij}$ de $\mathbf{M}^k$(G) (electrones que comparte el átomo '*i*' con el átomo '*j*') entre el numero de electrones que el átomo '*i*' comparte con todos los demás átomos en la molécula a *k*-distancias, incluido el mismo [Esto es igual al grado del vértice de orden *k*, ${}^k\delta_i$, el cual puede ser calculado utilizando el operador suma sobre las k^{th} potencia de la matriz **M**(G)]. Los elementos ${}^k s_{ij}$ se definen a continuación:

$$ {}^k s_{ij} = \frac{{}^k a_i}{\sum_{i=1}^{n} {}^k a_i} = \frac{{}^k a_{ij}}{{}^k \delta_i} \quad \textbf{(1.4.3.2.3.1)} $$

Teniendo en cuenta lo planteado anteriormente, fue definida recientemente tres nuevas familia de índices topológicos (topo-químico) a partir de la aplicación de conceptos de la matemática discreta y el álgebra lineal a la química. Estos descriptores están basados en el cálculo de formas cuadráticas, lineales y bilineales y han sido aplicados en diversos estudios QSAR/QSPR y se obtuvieron resultados satisfactorios. Sin embargo, no siempre estos índices muestran un desempeño totalmente satisfactorio para la predicción de ciertas propiedades. De hecho no se puede esperar que un conjunto específico de índices sea superior absolutamente a otros conjuntos posibles y/o pueda producir buenos resultados en todos los problemas. Como ejemplo de lo antes planteado tenemos el estudio realizado para predecir la permeabilidad de moléculas orgánicas y el trabajo referido a la predicción de estabilidad de péptidos y proteínas, entre otros.

1.5 Conclusiones parciales

Los métodos que emplean técnicas asistidas por ordenador usados en el descubrimiento, diseño, y optimización de compuestos con estructura y propiedades deseadas han desempeñado un rol importante en el desarrollo de fármacos que se encuentran actualmente en el mercado o en fase de estudios clínicos. Entre estos se destacan los modelos que relacionan cuantitativamente aspectos estructurales y propiedades de las moléculas (estudios QSAR). Numerosas bibliotecas para algoritmos genéticos y la química computacinal están disponibles para el desarrollador de nuevas aplicaciones. Entre ellas se destaca ECJ y CDK, ambas escritas en el lenguaje de programación Java, de código abierto, con una comunidad de usuarios y desarrolladores amplia y en pleno auge.

2 Diseño e Implementación

En este capítulo diseñaremos e implementaremos una GUI que permita explorar el espacio de los descriptores moleculares mediante algoritmos genético, para encontrar aquellos que permitan definir un modelo de predicción que se ajuste mejor a los datos. Pero para ello nos hace falta implementar la nueva familia de descriptores algebraicos de manera que mantenga la información topológica ya existente, nos hace falta diseñar e implementar una GUI para el Cálculo de esta nueva familia de descriptores y por último modelar el problema de exploración del espacio de los descriptores moleculares mediante el uso de Algoritmos Genéticos.

2.1 Implementación de los Descriptores Algebraicos

Extendemos la familia de descriptores algebraicos ya existente incorporándole:

PSA, Log P, Refractividad, Carga, dureza y suavidad. Todas son propiedades que a diferencia de las demás tienen en consideración el ambiente del átomo, o sea el vecindario de cada átomo y el tipo de átomo, o sea la hibridación así carbonos pueden existir 4 como sp3, sp2 alifático, sp2 aromático y sp3. Además de la hibridación que no es más que si es simple, doble o triple enlace se tomen consideración el ambiente de ese átomo. Para las propiedades anteriores hay una solo valor para cada átomo no importa quién es el ambiente ni si es doble tripe o simple enlace. Esa es la ventaja de cada uno de ellos. De esas nuevas, las 3 últimas necesitan ver a molécula en 3d, si se le da la molécula en 2d no dan un valor por eso también es buena o al menos diferente porque tienen en consideración todo lo anterior peor también geométricamente.

PM y DS dos nuevas matrices operadoras. La de probabilidad mutua la suma de todos los elementos suma 1 peor ni la suma de las filas ni la suma de, las columnas suman 1, lo cual si hace la SS. En el caso de PM no es más que la probabilidad de dos átomos comparte electrones en la molécula que forman. En el caso de la DS es una matriz que sus filas o sus columnas ambas suman 1. La suma de todos los elementos de la matriz sumarian el número de átomos de la molécula. La SS antigua solo las filas suma 1 las columnas no. De hecho esa es la diferencia entre una matriz estocástica simple y una doble estocástica que la doble suma 1 tanto por filas como por columnas. (Philip A. K.K.)

```
function [c, r] = sk(G, tol, g)
[n, n] = size(G);
r = ones(n,1); c = r;
d = G'*r + g*sum(r);
while norm(c.*d - 1,1) > tol
    c = 1./d;
    r = 1./(G*c+ g*sum(c));
    d = G'*r + g*sum(r);
end
```

Fig 2.1.1 Psedocodigo del balanceo de la matrix

Los trucajes en la matriz todos son algo nuevo nunca se ha publicado, hasta ahora solo se usaba la matriz completa sin truncar ningún valor. Los 3 filtros posibles usados, auto retorno, no auto retorno y lag k todos son nuevos.

3 nuevos grupos locales que son a) metilos o C terminales, b) átomos alifáticos y c) átomos aromáticos. Los dos sistemas fundaméntales de la química es el alifático y el otro aromático por eso calcular índices sobre esta porciones de la molécula, se comportan diferente, muy diferentes. Los metilos terminales también son diferentes átomos de los demás C átomos. Entre más existan más posibilidades de rotación tienen la molécula y más biodegradable y promiscua será la molécula o sea es inespecífica y actúan en muchos sitios lo cual es malo, porque aumenta la toxicidad.

El uso de invariantes de norma, média, estadística y clásicas además con la posibilidad de estandarizar con la deviación estándar y a media o no.

Para la implementación de la extensión de la familia de descriptores algebraicos partiremos de una biblioteca gratuita, de código abierto y disponible en Java, orientada fundamentalmente para la química informática y la bioinformática, nos referimos a CDK (STEINBECK, HAN et al. 2003) (STEINBECK, HAN et al. 2003), para representar tanto el comportamiento de las estructuras moleculares, como las transformaciones que se aplicarán sobre ellas.

Los códigos fuentes en Java están organizados mediante colecciones de clases nombradas, denominadas paquetes, estos por lo general presentan un estilo propio en sus nombres que se corresponde, en cierta forma, con direcciones de Internet invertidas (STEINBECK, HAN et al. 2003). Desde que la biblioteca CDK forma parte del proyecto OpenScience (Project) mantiene una disposición de paquetes para sus códigos fuentes encabezada por org.openscience.cdk.

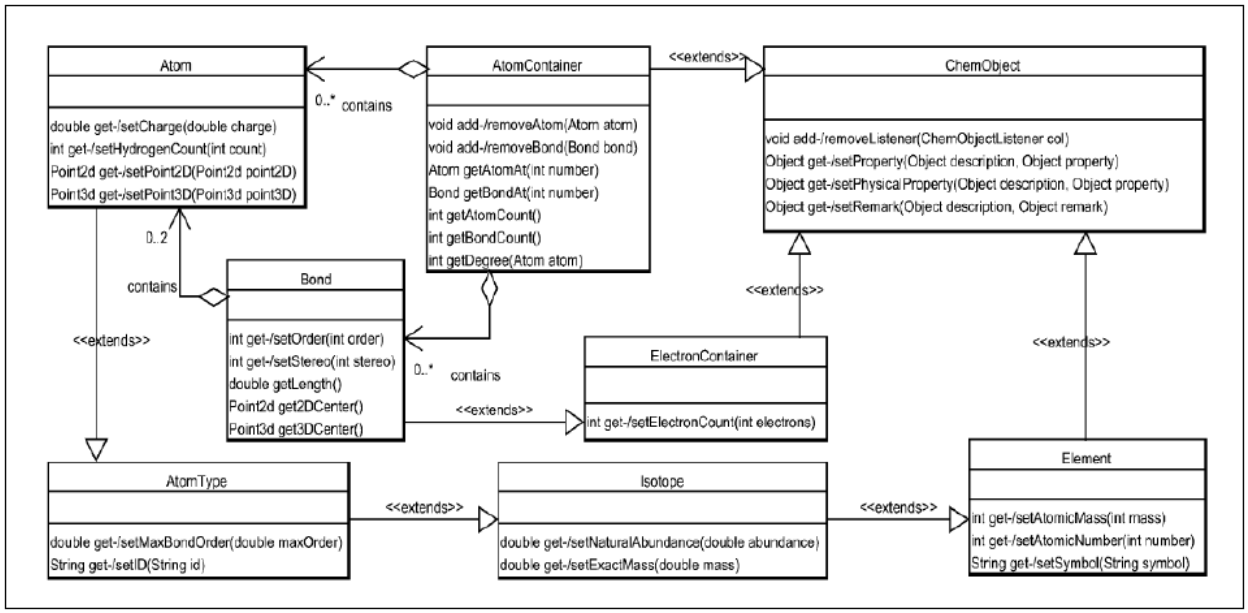


Figura 2.1: Diagrama UML con las principales dependencias entre clases en CDK

Las clases contenidas en la sección raíz de la jerarquía de paquetes son representaciones formales de conceptos básicos de química, como átomos, enlaces, moléculas, etc. La figura 2.1 muestra un diagrama en UML con las relaciones y dependencias entre las clases fundamentales de CDK (STEINBECK, HAN et al. 2003), donde podemos apreciar la clase ChemObject que se presenta como la superclase medular. Este diseño es lógico desde el punto de vista químico y sirve como base de un mecanismo simple para la creación de átomos y moléculas.

Las clases orientadas al cálculo de índices moleculares están agrupadas en el paquete org.openscience.cdk.qsar.descriptors.molecular y poseen entradas de especificación definidas mediante OWL (BECHHOFER, HARMELEN et al. 2009) un lenguaje de representación de conocimiento (BECHHOFER, HARMELEN et al. 2009), ubicadas en el fichero descriptor-algorithms.owl dentro del paquete org.openscience.cdk.dict.data con el objetivo de coordinar un identificador unívoco para cada descriptor implementado.

Para desarrollar la implementación de los descriptores algebraicos en el entorno de trabajo de CDK se propone hacer uso de la flexibilidad que nos brinda el lenguaje Java en los espacios de

nombres de las clases, para incorporar nuevas clases en paquetes ya existentes, mediante archivos independientes.

Para añadir un descriptor algebraico que cumpla con los requisitos expuestos tenemos que crear en el paquete org.openscience.cdk.qsar.descriptors.molecular un subpaquete que agrupará la colección de clases para los nuevos índices, definido como org.openscience.cdk.qsar.descriptors.molecular.algebraic. Este paquete representará el comportamiento de los índices algebraicos dentro de CDK, implementamos de la clase IMolecularDescriptor, que asume el esquema principal de un descriptor molecular, para conformar las clases BilinearDescriptor, QuadraticDescriptor y LinearDescriptor, como se ilustra en la figura 2.2.

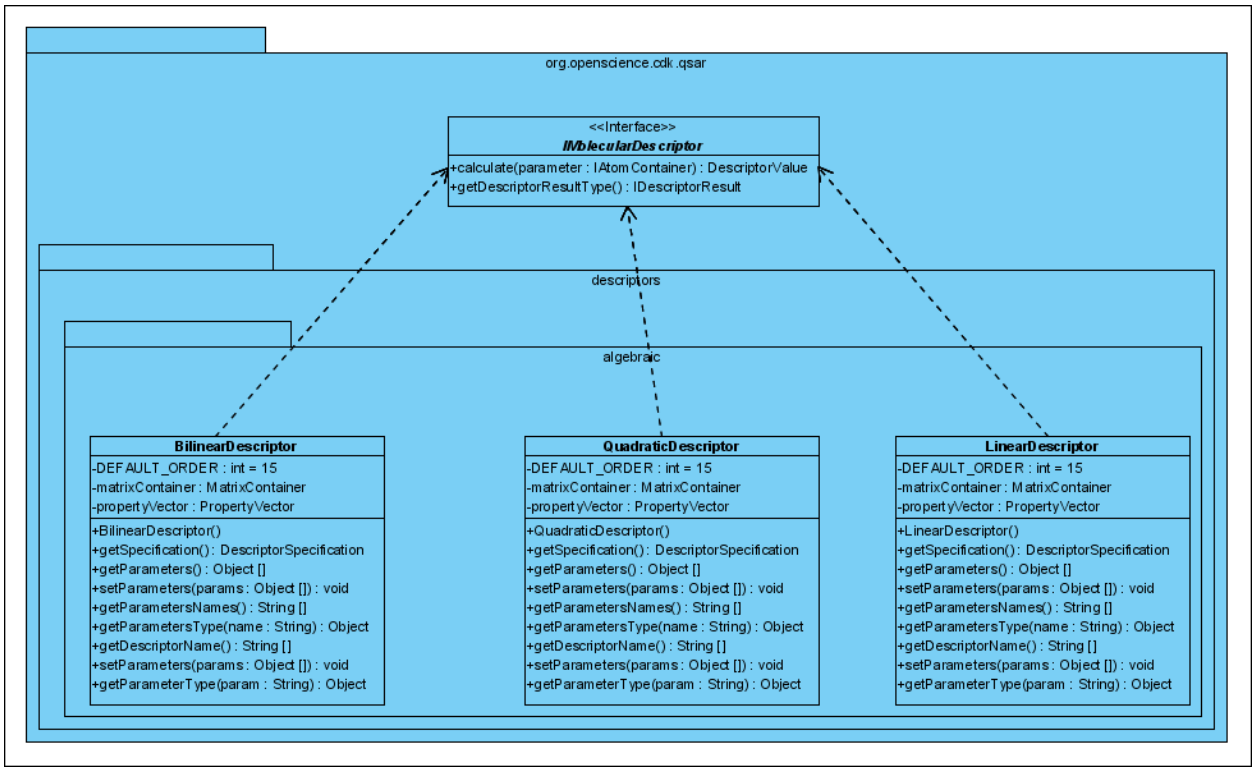


Figura 2.2: Diagrama de clases para los descriptores algebraicos

Para completar el esquema del cálculo en forma algebraica de índices moleculares, introducimos un conjunto de clases representativas para cada una de las posibles ponderaciones atómicas (entiéndase, cada propiedad define una clase) y para cada una de los posibles procedimientos de construir la matriz, lo que se cumple de manera similar para las demás formas algebraicas.

Por ejemplo tomemos una jerarquía de herencias para un descriptor bilinear (BilinearDescriptor) formado por matrices no estocásticas (NonStochasticBilinearDescriptor) que heredan de la clase BilinearDescriptor, para todos lo átomos o sea Total, que utiliza como propiedad atómica la hardness y electronegatividad de Pauling (Hardness_Electronegativity. Consecuente a la definición de este nuevo descriptor definido por la clase Hardness_ElectronegativityTotalNonStochasticBilinearDescriptor, incluimos la declaración en el lenguaje OWL que le corresponde, ver figuras 2.3 y 2.4 respectivamente.

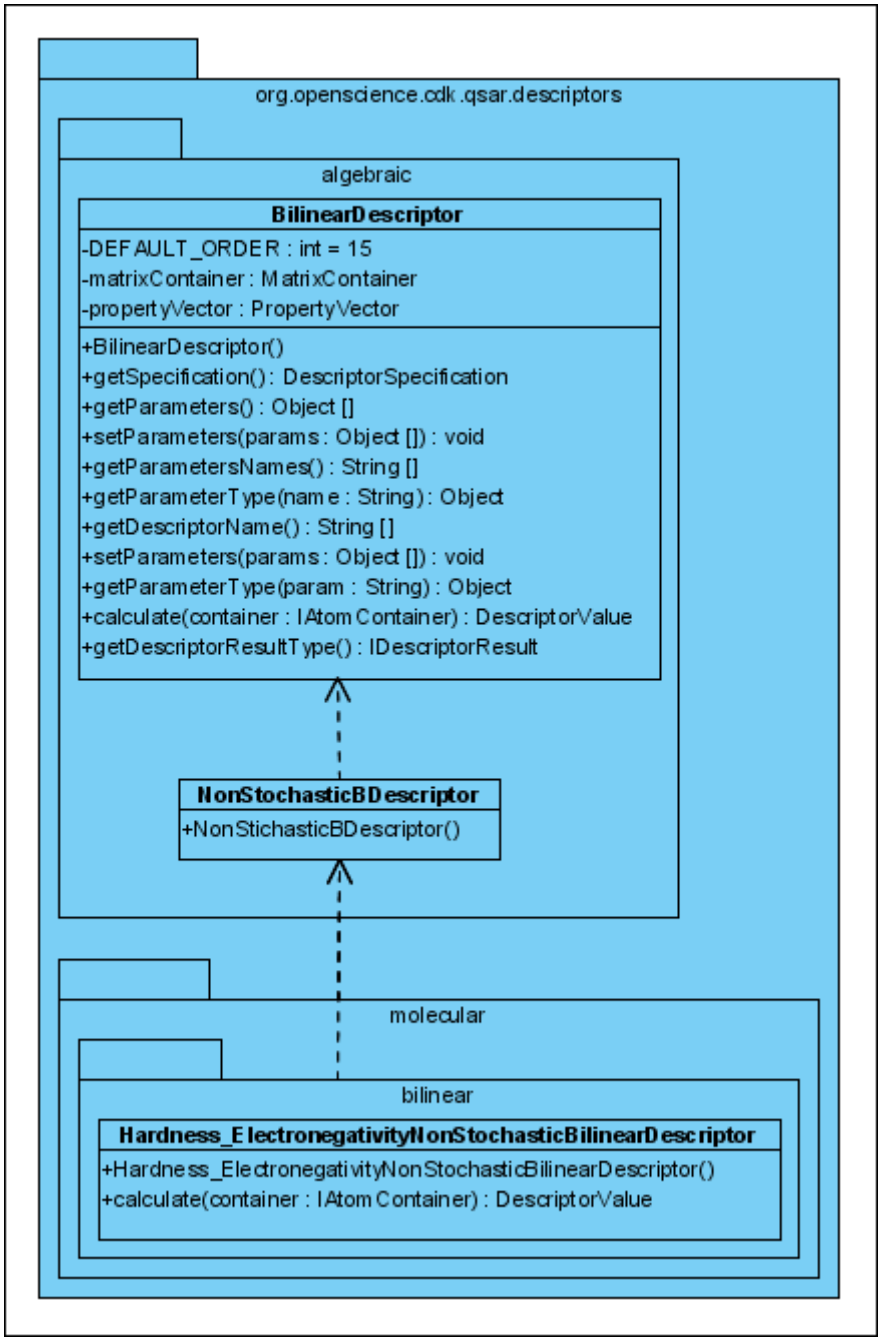


Figura 2.3 Herencia de clases para un descriptor lineal no estocástico ponderado con electronegatividad

```
<Descriptor rdf:ID="h_ensbilinear">
	<rdfs:label>
		BilinearFormNonStochastic(Hardness_Electronegativity)
	</rdfs:label>
	<dc:contributor rdf:resource="#cysva"/>
	<dc:date>2011-04-14</dc:date>
	<definition rdf:parseType='Literal'>
		The BilinearNonSthochastic descriptor using atomic weight
	</definition>
	<isClassifiedAs rdf:resource="#algebraicFormDescriptor"/>
	<isClassifiedAs rdf:resource="#molecularDescriptor"/>
</Descriptor>
```

Figura 2.4 Definición en OWL para el descriptor de la figura 2.3

Con el objetivo de facilitar el proceso de incorporar un nuevo descriptor molecular para formas algebraicas completamente funcional y compatible con los requisitos de CDK, se diseñó e implementó una colección de clases que representa todas las posible alternativas, aislando solo dos pasos para satisfacer esta tarea, los que se exponen a continuación:

1. Crear una clase para especificar la propiedad que se desea evaluar en el descriptor y asignarla al vector de propiedad, esta clase tendrá como superclase una representación de su configuración sea el descriptor bilineal, lineal o cuadrático y al método de construir la matriz.

2. Definir la especificación del nuevo descriptor en el lenguaje (OWL) e incorporarla al fichero descriptor-algorithms.owl.

En el anexo 2 se expone una relación de los nombres para los descriptores algebraicos implementados.

2.2 GUI para el Cálculo de descriptores (TOMOCOMD-CARDD)

2.2.1 Diseño

Presentamos la aplicación TOMOCOMD-CARDD como extensión del proyecto desarrollado por Rajarshi Guha disponible en su página Web (GUHA 2010), esta permite calcular de forma sencilla muchas de las familias de descriptores ya existentes implementados en las bibliotecas CDK (STEINBECK, HAN et al. 2003) y JOELib (WEGNER 2004), todos los descriptores serán

evaluados en conjunto o se puede determinar una configuración específica seleccionando un descriptor individual. Entre sus principales características encontramos:

• Detecta automáticamente las clases definidas en el diccionario de descriptores en el CDK

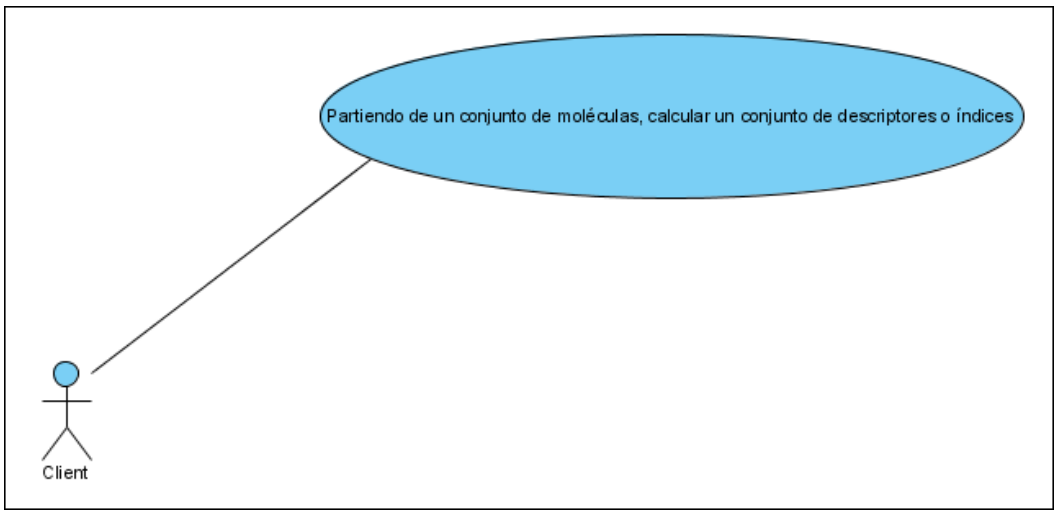
• Se pueden seleccionar los descriptores individualmente o por categorías.

• Compatible con los formatos de entrada SDF, SMI, MOL o MDL de representación de moléculas.

• El archivo de salida se puede generar en texto plano delimitado, o en ARFF.

Estos son cargados mediante la arquitectura de plugins que esta aplicación presenta, con solo agregar en el classpath el archivo que contiene los descriptores compatibles con el diseño estructural de la biblioteca (STEINBECK, HAN et al. 2003), se incorporan los descriptores algebraicos, desde un archivo .jar independiente, los que nos proporciona un apropiado esquema el caso que se desee extender las funcionalidades o incrementar el número de descriptores algebraicos.

Abajo presentaremos el diagrama de casos de uso y actor.



2.2.2 Implementación

La GUI fue implementada en Java manteniendo todo lo de la aplicación de BMD de (GUHA 2010) y lo se agregó el panel “QuBiLs-MAS 2D” para la configuración de los descriptores moleculares algebraicos (TOMOCOMD-CARDD), para más detalles ver Anexo.

2.3 GUI para optimización de modelos con índices idóneos o optimizados (GENOM-FLEXD)

Para el diseño e implementación de la GUI para exploración del espacio de busqueda de los descriptores, primeramente partimos de una biblioteca libre ECJ, agregándole dos clases nuevas DescriptorSpecie heredando de Specie y DescriptorIndividual heredando de VectorIndividual y algunos métodos para que fuera posible modelar nuestro problema en Algoritmos Genéticos, para extender el ECJ pues utilizaremos una variante no tradicional del Algoritmo Genetico. Como muestra el diagrama 2.3.1.

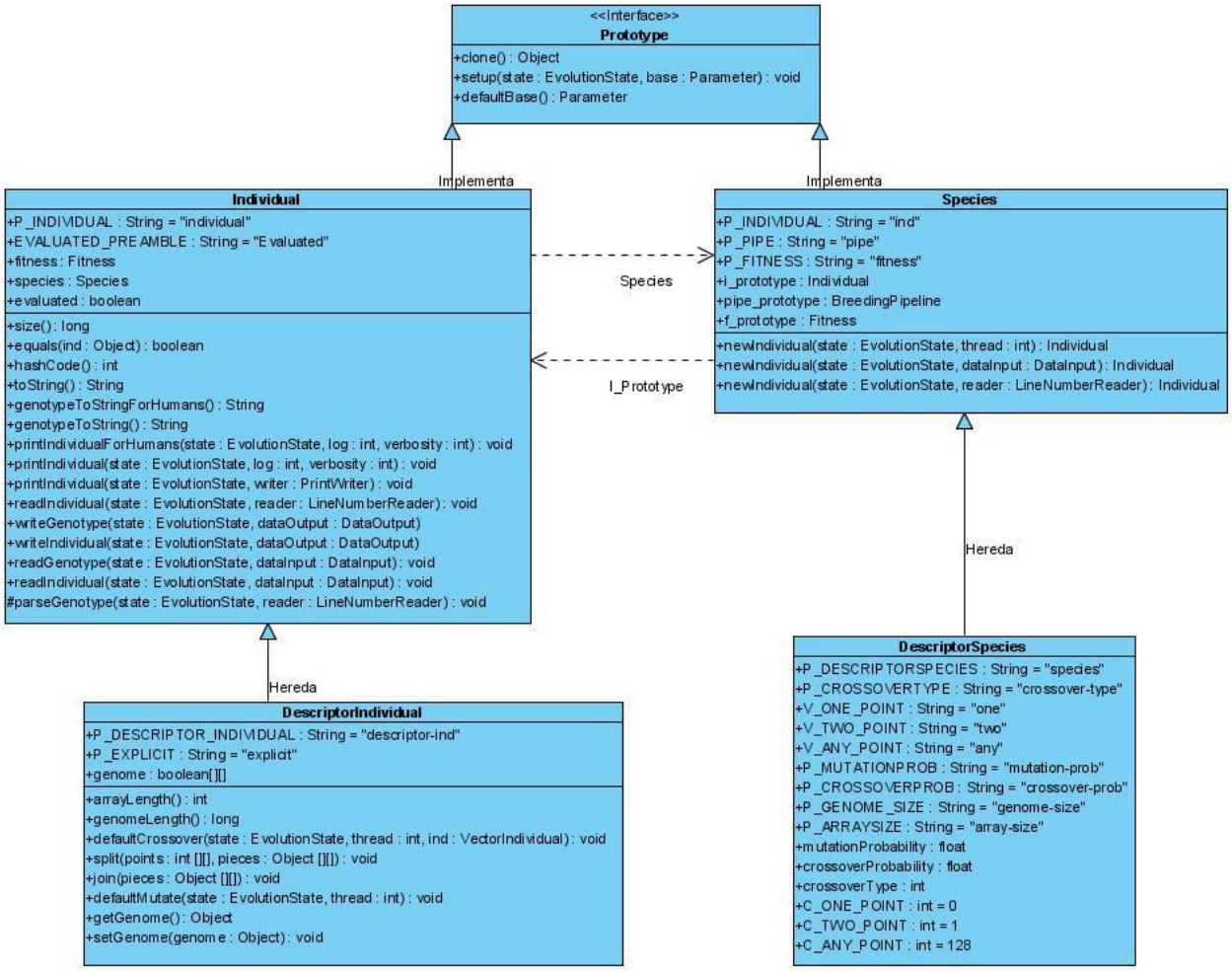
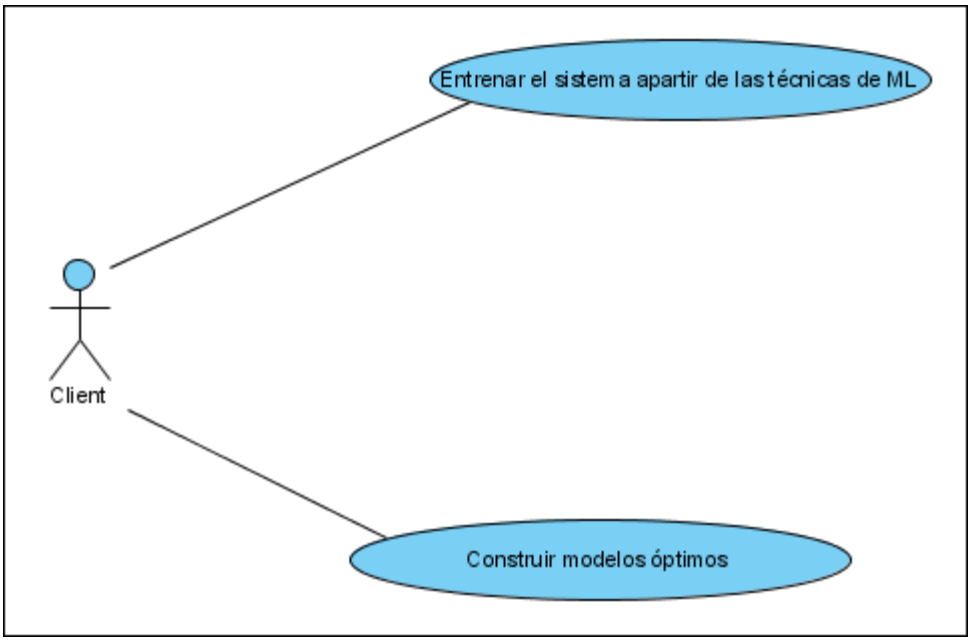


Diagrama 2.3.1: Clase DescriptorSpecies y DescriptorIndividual

Tambien presentaremos el diagrama de casos de uso y actor.



2.3.1 Mecanismo de detención de soluciones no factibles

encodeGenome(descriptorName : String): String

Algoritmo 2.3.1.1: Encode a Genome code for a descriptor name

decodeGenome(descriptorCode : String): String

Algoritmo 2.3.1.2: Decode a Genome code for a descriptor name

evaluateGenome(descriptorCode :String): boolean

Algoritmo 2.3.1.3: check a genome code, if is a valid genome return true else false

2.3.2 Modelación del Problema con Algoritmos Genéticos

Para la exploración del espacio de soluciones de los descriptores moleculares algebraicos usaremos la siguiente codificación, en un cromosoma hay n genes y en cada gen hay 35 bits, donde:

- 2 bits para representar 3 formas algebraicas, Bilineal, Cuadrática y Lineal.
- 6 bits para representar 64 formas matriciales y sus respectivos órdenes, DoubleStochastic K, NonStochastic K, MutualProbability K y SimpleStochastic K, donde K puede variar de 0 a 15.

- 3 bits para representar 8 grupos, Total, AAtoms, CAtoms, DAtoms, GAtoms, MAtoms, PAtoms y XAtoms.
- 8 bits para representar 228 trucajes, ALL, WRS, NWRS, LAG K, donde K puede variar de 1 a 15, mas los intervalos 1-2,…,1-15,2-2,…,2-15,…14-15.
- 6 bits para representar 55 propiedades.
- 10 bits para representar 989 invariantes en total de Norma, Média, Estadisticas y Clásicas con K máximo de 15.

La figura 2.3.2.1 muestra un ejemplo de un cromosoma de n=4 genes.

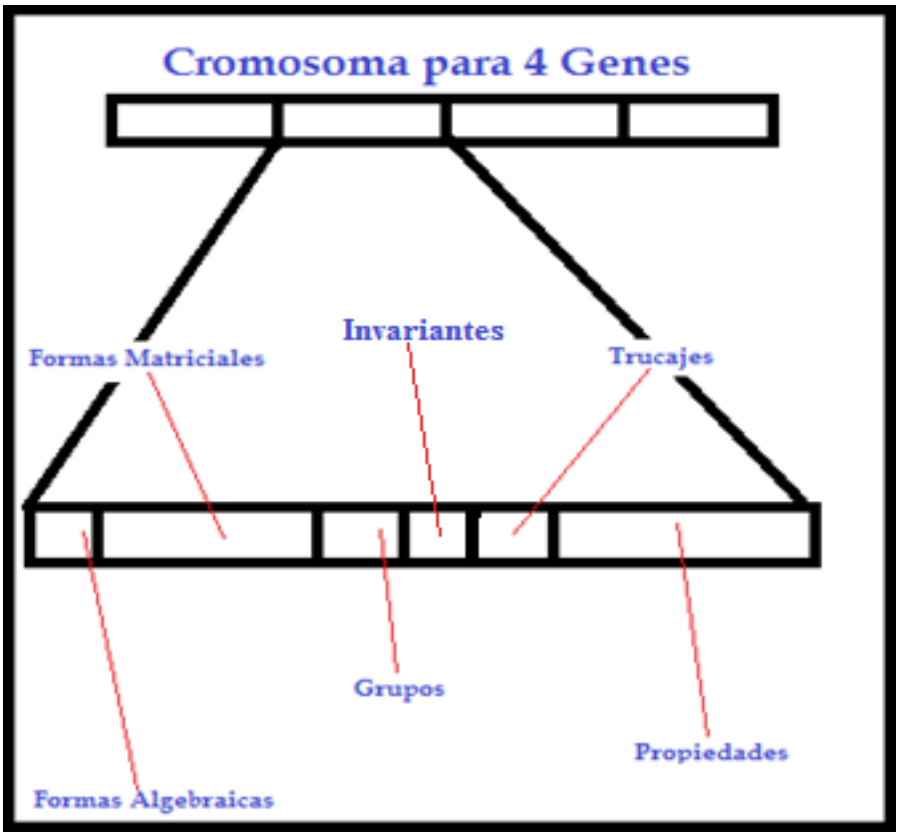


Figura 2.3.2.1: Ejemplo de un cromosoma con n=4 genes.

El parámetro de número de genes en un cromosoma será controlado en la aplicación.

2.3.2.1 Individuo

El diseño que comprende la biblioteca evolutiva ECJ para codificar los individuos, deja bien claro las posibilidades de extenderla hacia otros entornos más complejos.

Nuestro individuo será una clase DescriptorIndividual que hereda de VectorIndividual donde definimos el genoma como un boolean[][], que a su vez sus características están en la clase

QSARModelSpecies que hereda de Species. O sea para cada i de 0 a n en boolean[i] representará un descriptor y para cada j de 0 a 34 en boolean[i][j] representará una característica del descriptor i.

2.3.2.2 Especie

Una colección de individuos descriptores con características similares, define un espacio de información denominado "especie", donde comparten conocimientos y ajustan parámetros que controlan el algoritmo genético. Una vez creado el paquete ec.algebraic incorporamos la clase DescriptorSpecies favorable a garantizar la perpetuación de la especie durante la supervivencia, en esta clase podemos encontrar:

- Las probabilidades de mutación y cruce para cada operador,
- El tipo de cruzamiento,
- El modelo que se usará para construir la función de evaluación, con sus respectivos datos de aprendizaje y
- El archivo con las moléculas a sembrar en la población inicial, entre otros.

2.3.2.3 Población Inicial

Se genera aleatoriamente la población inicial a partir de un conjunto de tamaño definido por el usuario en la aplicación constituido por cromosomas con características predeterminadas, es decir, se siembra una cantidad fija de descriptores con cualidades y efectos conocidos, que representan posibles soluciones parciales del problema, para completar el número de individuos requerido en el proceso evolutivo se propone incrementar este conjunto inicial establecido, aplicando transformaciones en sus ejemplares mediante el operador de mutación (RandomGenerator), detallado en la sección 2.3.2.5.

Es importante garantizar que la población base inicial, mantenga diversidad estructural sobre el espacio de soluciones, para que lo represente significativamente y evite la convergencia prematura.

2.3.2.4 Operador de Cruzamiento

Para el nuestro problema usaremos el cruazamiento de un punto, dos puntos y cualquier punto, estos serán a nivel de cromosoma, o sea para el caso de un punto se hace referencia a un

gene o descriptor en el cromosoma como muestra en la figura 2.3.2.4.1 un cromosoma P1 constituido por 4 genes o descriptores y otro cromosoma P2 constituido por 4 genes o descriptores, el resultante será dos progenitores que serán dos modelos donde se han intercambiado sus descriptores o genes. Obviamente este operador no producirá ningún tipo de problema, o sea todas sus soluciones serán factibles.

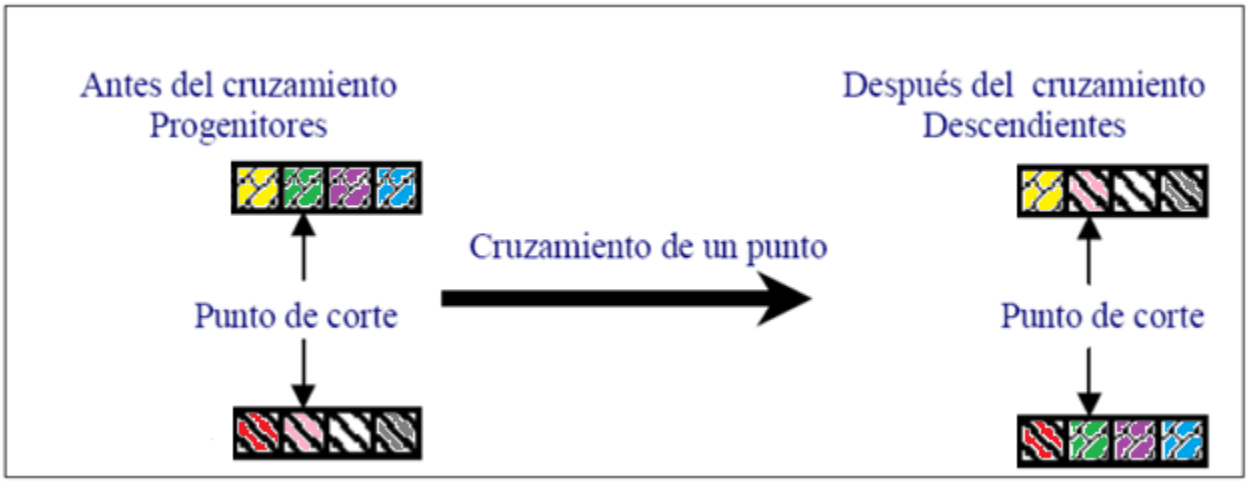


Figura 2.3.2.4.1: El operador de cruzamiento para dos cromosomas.

2.3.2.5 Operador de Mutación

El operador de mutación que se usará en nuestro problema será a nivel de gene como muestra la figura 2.3.2.5.1. Donde el punto de mutación referencia una posición en el gene con un rango de valor entre 0 a 35. Este operador si introducirá soluciones no factibles al conjunto solución, pero el algoritmo 2.3.1.3 solucionará el problema de las soluciones no factibles.

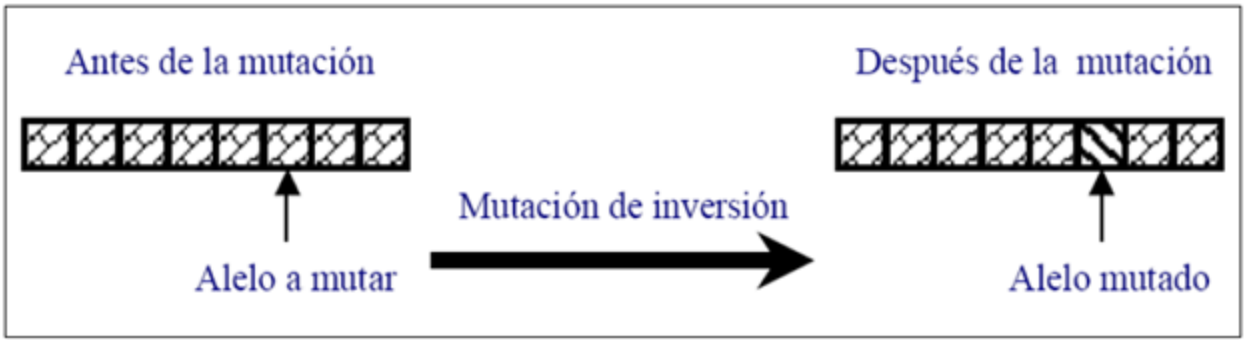


Figura 2.3.2.5.1: El operador de mutación sobre un gene o descriptor.

2.3.2.6 Operador de Selección

Los operadores de selección que se usará serán Random Selection y Tournament Selection, donde las combinaciones de descriptores o modelos QSAR compiten entre si y los mejor valores dan su función de fitness serán seleccionados, como muestra la figura 2.3.2.6.1.

Figura 2.3.2.6.1: Operador de Selección por Torneo

2.3.2.7 Función de Fitness

La función que se propone en este trabajo es compleja, los datos obtenidos formarán una instancia que será estimada por un modelo creado con técnicas de aprendizaje automático en Weka. El procedimiento de evaluar un modelo se definirse como:

1. Verificar que el individuo es un QSARModelIndividual y no fue evaluado previamente.
2. Decodificar a cada gen del cromosoma en en cada descriptor algebraico respectivamente.
3. Calcular respectivamente cada descriptor algebraico.
4. Construir una instancia de Weka con estos datos, construindo el modelo QSAR.
5. Evaluar el modelo en Weka usando Split con un 66%, 10 fold cross-validation o compararlo con un modelo QSAR externo de manera de concebir ajuste y calidad.
6. Asignar el valor conseguido como fitness para este individuo.
7. Marcar el individuo como ya evaluado y seguir a iteración.

2.3.3 Diseño de la GUI

El diseño de la aplicación se basa en la plataforma Swing Application de Java. La aplicación está formada por 5 paneles de tabulación ordenados según el orden de funcionalidad de la misma. Los 4 primero servirán de configuración de parámetros, el último inicializará el

proceso de creación del modelo con Weka, evaluando los modelos en ECJ, como la exploración del espacio de búsqueda de los descriptores moleculares algebraicos usando ECJ. El archivo de parámetros para la plataforma ECJ de los Algoritmos Geneticos de creará dinámicamente, o sea según las especificaciones del usuario y pueden referir a los detalles de implementación en el Capitulo 3..

2.4 Conclusiones parciales

En este trabajo logramos:

1. Definir e implementar una nueva familia de descriptores moleculares algebraico basados en la teoría de grafos y algebra lineal de manera tal que amplíen los ya existentes considerando la información topológica y química no incluida en los descriptores actuales.
2. Diseñar e implementar una GUI que permite el cálculo de los distintos descriptores algebraicos a un grupo diverso de moléculas.
3. Modelar el problema de selección de descriptores moleculares mediante algoritmos genéticos.
4. Diseñar e implementar una GUI que permite explorar el espacio de los descriptores moleculares algebraicos mediante algoritmos genéticos para encontrar sus combinaciones óptimas que permiten definir modelos de predicción que mejor se ajuste a los datos.
5. Validar y verificar la herramienta computacional.

3 Herramienta Computacional: GENOM-FLEXD

El resultado de nuestro trabajo es la herramienta computacional GENOM-FLEXD. Esta tiene como componentes ECJ para el uso de los Algoritmos Genéticos, WEKA para el uso del Aprendizaje Automatizado y ECDK para construir el espacio de soluciones de los descriptores moleculares algebraicos que serán explorados por los algoritmos geneticos, luego partiendo de la entrada de un data set se la aplicación efectuará varios procesos hasta la obtención de una salida con los modelos óptimos. La figura 3.1.1 brindanos una visión general del funcionamiento de la aplicación.

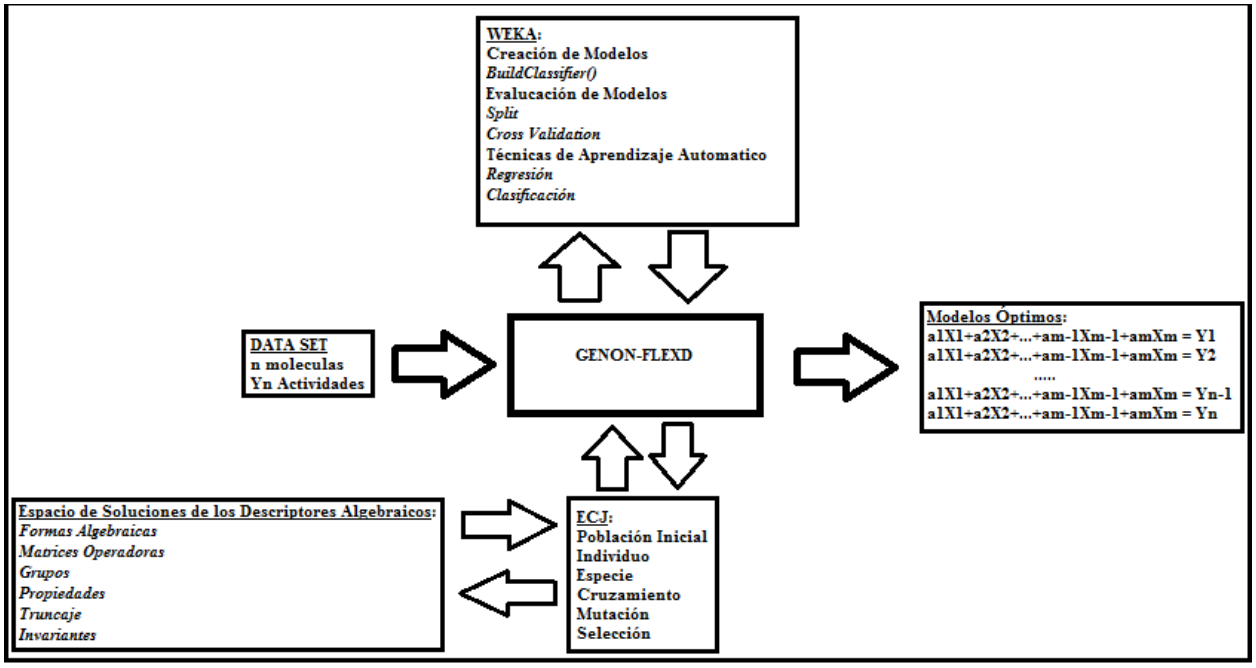


Figura 3.1.1: Esquema General del Funcionamiento de la Aplicación GENOM-FLEXD

3.1 Data Set: Moléculas y Parametros Y

El data set está formado por las moleculas y sus actividades, primeramente el usuario debe introducir un conjunto de archivos que contiene las moléculas que le llamaremos de "*molecule*

files", posteriormente el usuario puede introducir otro archivo donde están las Y de estas moléculas que llameremos de "*Y Parameter file"* la cantidad de moléculas tiene que ser igual al númerod de Y existentes, o introducir manualmente la Y respectiva de cada molécula, usando el botón "*Insert"*. La componente "viewer" como el nombre lo dice es el visualizador de las Y, luego al seleccionar una Y en la lista "Available Y Paremeters" su contenido aparecerá en la parte de abajo "Y Parameter Content", donde el usuario puede modificar los valores contenidos de un Y Parameter usando el botón "Apply Changes". Despues de tener todas las Y Parameters en la lista "Available Y Parameters", el usuario es libre de seleccionar cuales de estas Y disponibles pretende usar para la contrucción de su Modelo. La figura 3.1.1.1 nos dá una visión grafica de la aplicación.

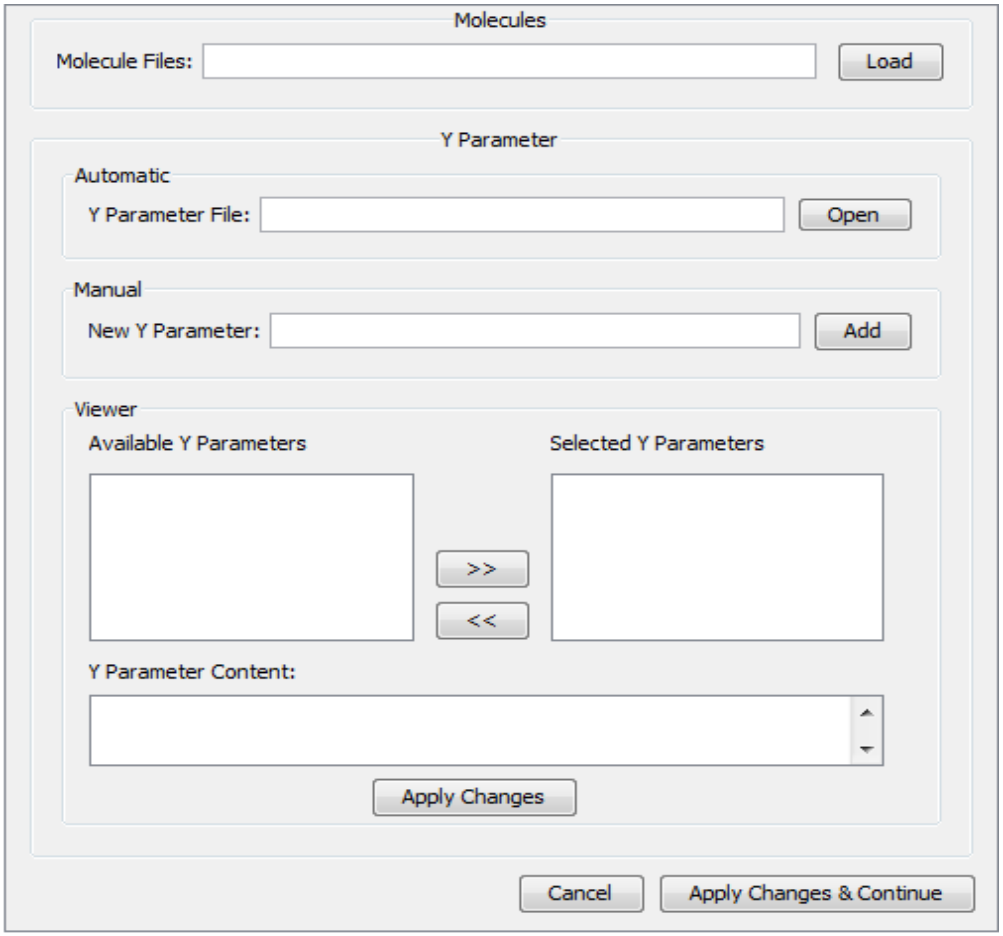


Figura 3.1.1.1: Proceso de introducción del data set.

3.2 ECJ: Parametros

Después de haber seleccionado el dataset partimos para la definición de los parámetros de la exploración de estos espacios usando los Algoritmos Geneticos en la plataforma ECJ.

Los parametros de la plataforma ECJ se definen en forma de archivo, para nuestro problema se definirán un archivo *"genom-flexd.params"*, donde se definirán el tamaño de la población inicial, la cantidad de generaciones, el número de descriptores en un cromosoma, el tipo y la probabilidad del cruzamiento, la probabilidad de mutación y el tipo de selección a usar. Aquí no se presentará la función de fitness pues esta será creada por WEKA apos iniciar el proceso, hablaremos de ella más adelante. Luego el archivo de parámetros del ECJ de nuestra aplicación terá un proceso de creación diferente de las demás aplicación porque este será creado y definido por el usuario de manera dinámica.

Como ejemplo tenemos para una población inicial de 10 cromosomas, una cantidad de generación de 20, teniendo 4 descriptores en cada cromosoma, el tipo de cruzamiento de un punto con una probabilidad de 1.0, una probabilidad de mutación de 0.01 y el tipo de selección de Tournament Selection, como muestra la figura 3.1.2.1.

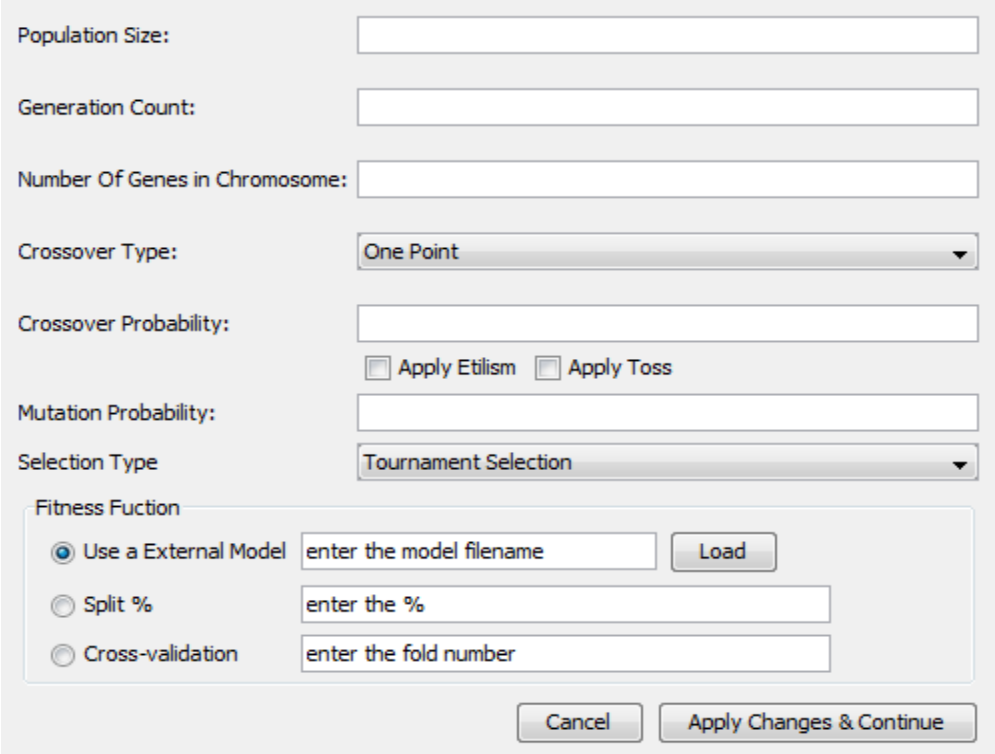


Figura 3.1.2.1: Proceso de creación e definición del archivo de parámetro del ECJ

3.3 Espacio de Solucciones: TOMOCOMD-CARDD

Apos haber seleccionado los parámetros del GA sobre la plataforma ECJ, partimos para definición del espacio de soluciones que será explorada por los algoritmos genéticos.

Para la construcción de modelos óptimos necesitamos buscar las X óptimas para estos modelos, estas X por otras palabras son la nueva familia de descriptores moleculares algebraicos implementados. El espacio de solución de estos descriptores será definido por el usuario, de manera tal que el mismo pueda definir cuales son los descriptores que pretende usar y posteriormente será explorado por los algortimos genéticos.

Como ejemplo, se puede definir un espacio de soluciones usando la forma algebraica *"Linear"*, la matriz operadora *"Non Stochastic"*, el grupo *"Total"*, las propiedades de *"Masa"* y *"Carga"*, el truncaje de *"All"* y como invariantes *"N1 – Manhatta Distance"* y *"G – Geometric Mean"* posteriormente usar el botón *"Apply Changes"* para definir los descriptores a usar según estas opciones que fueron seleccionadas, luego como resultado el espacio de soluciones posibles será acotado, sendo así el algoritmo genético solo explorará las soluciones existentes en este espacio. La figura 3.1.3.1 nos ayudará a entender este proceso.

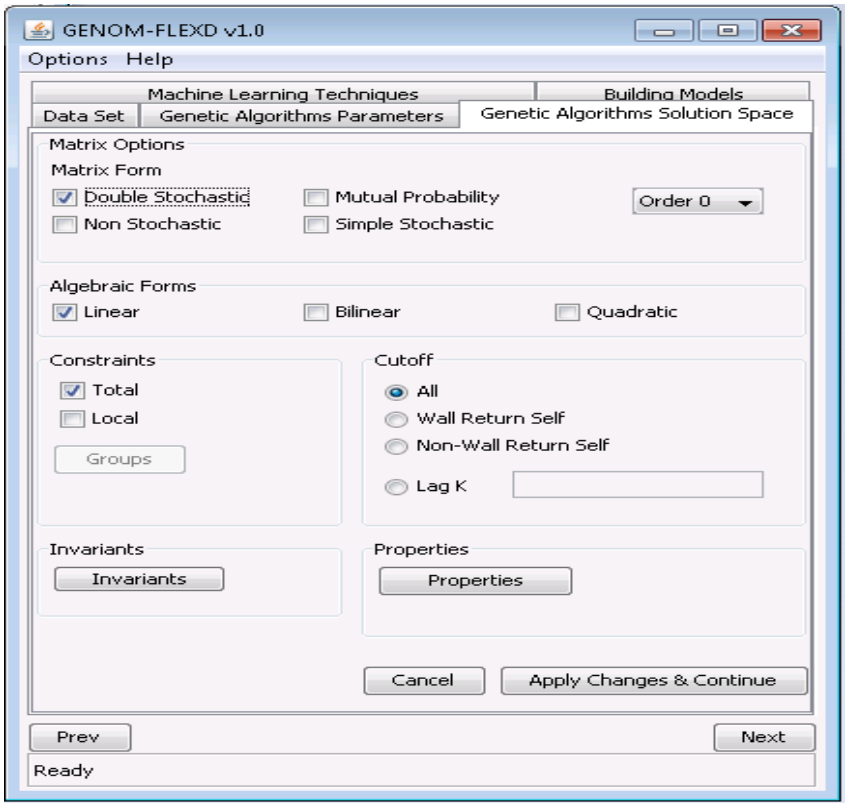


Figura 3.1.3.1: Proceso de definición del Espacio de Soluciones

3.4 WEKA: Técnicas del Aprendizaje Automatizado

Después de haber introducido el data set, las actividades o Y pueden tomar valores discretos o continuos. Unas de las componentes de nuestra herramienta es el suporte del aprendizaje

automatizado, para ello esta aplicación brindanos 4 técnicas de regresión para el manejo valores continuos y 4 técnicas de clasificación para el manejor de valores discretos de las actividades de las moléculas introducidos en el data set.

Como técnicas de regresión tenemos KNN, MLR, SVM y J48 y como técnicas de clasificación tenemos KNN, LR, SVM y ID3. El usuario solo podrá seleccionar una técnica de ambas, o sea una de regresión y una de clasificación. En caso de que los valores no sean discretos se efectuará una discretización del los valores continuos. La figura 3.1.4.1 nos ayudará a entender este proceso.

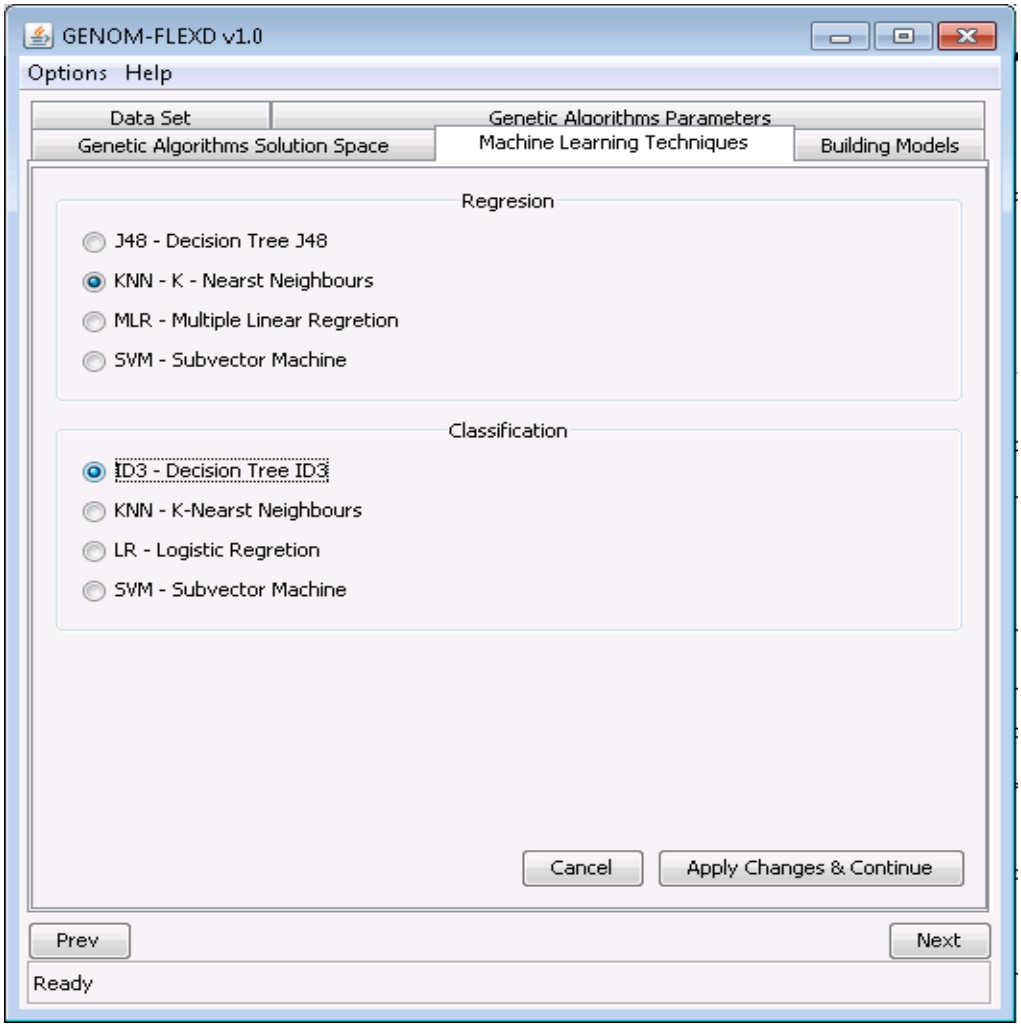


Figura 3.1.4.1Proceso de selección de las Técnicas a usar en el Aprendizaje Automatizado

3.5 Construcion de Modelos

Após haber configurado todas las opciones descriptas en los epigrafos 3.1.1 hasta el 3.1.4, inicializemos el proceso de construcción y búsqueda de modelos. Abajo tenemos un psedocodigo del funcionamento de este proceso.

1. Se construirá usando Weka un modelo QSAR con la función BuildClassifier con los datos incluidos en el data set y los parámetros definidos para el ECJ.
2. Se evaluará el modelo usando Weka con las funciones Split 66% con el objectivo de lograr velocidad, o la función 10 fold cross validation con el objectivo de lograr ayuste, o usando el modelo QSAR externo, luego en este proceso se busca un balance entre ayuste y velocidad.
3. Se explorará otras soluciones factibles usando ECJ, según el espacio de soluciones y los parámetros definidos.
4. Se entrenará el modelo con las técnicas del Aprendizaje Automatizado y se vuelve al paso 2.
5. Este proceso se detendrá cuando halla encontrado el modelo óptimo o por opción del usuario en caso de que lleve días y no ha visto ningún progreso.

La figura 3.1.6.1 muestra el proceso de ejecución para una corrida.

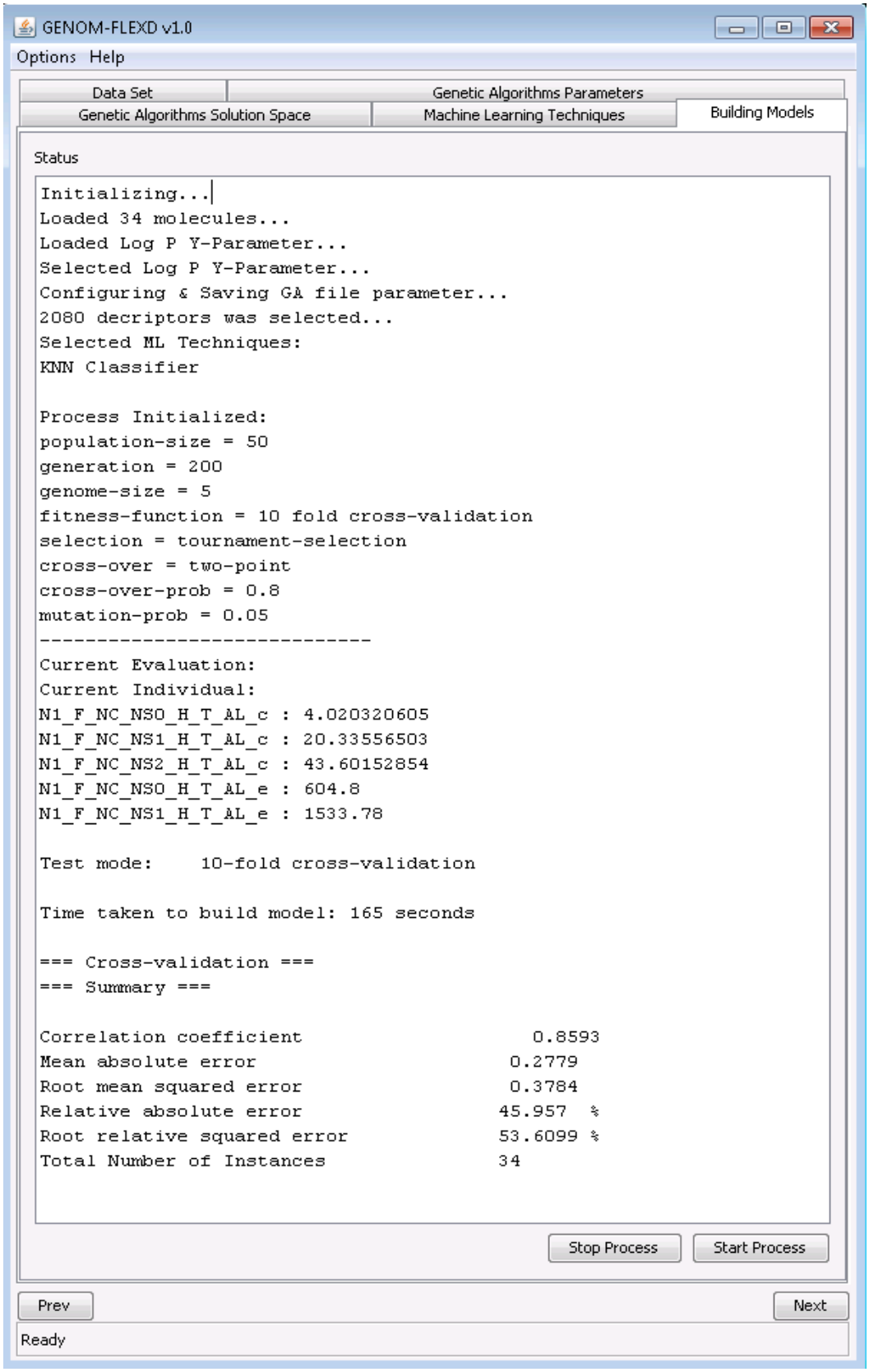


Figura 3.1.6.1:Proceso de explotación y creación de Modelo QSAR

La Aplicación GENOM-FLEXD como habíamos dicho esta sobre muchas plataformas, luego unas de las opciones que tiene es que el usuario pueda definir si desea o no trabajar con H. También tiene la posibilidad de definir el tipo de salida que quiere como tabla o graficas como se hace en WEKA. Por último la aplicación brinda al usuario la posibilidad de salvar o cargar todas las confuguraciones hecha y inclusive cargar un estado en las iteraciones del Algoritmo Genetico de manera que este puede dar seguimiento sin tener que empiezar desde el inicio.Como muestra la figura 3.1.6.2.

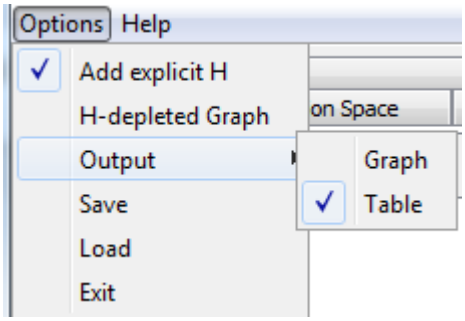


Figura 3.1.6.2: Menú General

Conclusiones

- Definimos e implementamos computacionalmente nuevos descriptores moleculares basados en la teoría de grafos y el álgebra lineal de manera tal que amplíen los ya existentes considerando la información topológica y química no incluida en los descriptores actuales.
- Diseñamos e implementamos una interfaz gráfica de usuario que permite calcular los distintos descriptores moleculares a un grupo diverso de moléculas.
- Modelamos el problema de selección de descriptores moleculares mediante algoritmos genéticos.
- Diseñamos e implementamos una interfaz gráfica de usuario que permite explorar el espacio de los descriptores moleculares algebraicos mediante el uso de los algoritmos genéticos para encontrar las combinaciones óptimas que permitan definir modelos de predicción que mejor se ajuste a los datos.
- Validamos y verificamos la herramienta mediante la utilización de conjuntos de datos internacionales.

Recomendaciones

1. Definir el operador de Migración en el Algoritmo Genetico.
2. Definir el operador Epidemy. Que significaría que una nueva generación es obtendida preservando un porcentaje de la población de los individuos y rellenado lo que falta de la población con indiviudo generados al azar.
3. Definir el operador Predatory. Que significa que los mejores individuos eliminan a los indiviudos similares a este o parecidos.
4. Incorporar métodos de validación más robusto como:

   Y-Scrambling

   Boostrapping
5. Someter la herramienta GENOME-FLEXD a un proceso de validación más exhaustivo.

Bibliografía

A Katrizky, U. M., V S Lobanov, M Karelson (2000). "Journal of Cheminformatics and Computer Science." 40.

Alba, E. and C. Cotta (2003). "Tutorial on evolution computation.".

ARENAS, M. G., L. FOUCART, et al. Computación Evolutiva en Java: JEO.

Axler, S. (1996). Linear Algebra Done Right. New York, Springer-Verlag.

BACK, T., D. B. FOGEL, et al. (1997). "Handbook of Evolutionary Computation." OUP/IOP.

BECHHOFER, S., F. V. HARMELEN, et al. (2009). "OWL Web Ontology Language W3C Semantic Web Activity.".

Bonchev, D. and D. H. Rouvray (1991). Chemical Graph Theory. Introduction and Fundamentals. New York, Abacus Press/ Gordon and Breach Science Publishers.

Browder, A. (1996). Mathematical Analysis. An Introduction. New York, Springer-Verlag.

Busacker, R. G. and T. Saaty (1965). Finite Graphs and Networks. New York, McGraw-Hill.

Castillo-Garit, J. A., Y. Marrero-Ponce, et al. (2006). "Atom-based 3D-chiral quadratic indices. Part 2: Prediction of the corticosteroid-binding globulinbinding affinity of the 31 benchmark steroids data set." Bioorg Med Chem **14**: 2398-2408.

CDL (2009) Chemical Descriptors Library (CDL).

Charton, M. (1996). Advances in Quantitative Structure-Property Relationships. Amsterdam, JAI Press.

ChemoJava (2009). "ChemoJava." from http://www.gstatic.com/.

CHEN, H. F. (2008). "Quantitative predictions of gas chromatography retention indexes with support vector machines, radial basis neural networks and multiple linear regresión." Analytica chimica acta.

CLARK, D. E. and D. R. WESTHEAD (1996). "J. Comput.-Aided Mol. Des." **13**: 516.

COSTA, J., N. LOPES, et al. "JDEAL, The Java Distributed Evolutionary Algorithms Library.".

DE JONG, K. A. and J. SARMA (1993) Generation Gaps Revisited. Foundations of Genetic Algotihms. **2**, 19-28

DEVILLERS, J. (1996). "Genetic Algorithms in Molecular Modelling." Academic Press, London.

Diudea, M. V. (2001). QSPR/QSAR Studies by Molecular Descriptors. Huntington, N.Y., Nova Science.

ECLAB (2010). from http://cs.gmu.edu/~eclab/projects/ecj/.

Edwards, C. H. and D. E. Penney (1988). Elementary Linear Algebra. New Jersey, USA, Prentice-Hall, Englewood Cliffs.

Estrada, E. and E. Molina (2001). "3D Conectivity Indices in QSPR/QSAR Studies." J. Chem. Inf. Comput. Sci. **41**: 791-797.

GOLDBERG, D. E., Ed. (1989). Genetic Algorithms in Search Optimization and Machine Learning. Reading, MA, Addison-Wesley.

GOLDBERG, D. E., K. DEB, et al. (1991) Genetic Algorithms, Noise, and the Sizing of Populations. IlliGAL Report No. 91010.

GUHA, R. (2010). "BMD Calculator." from http://rguha.net/code/java/cdkdesc.html.

Holland, J. (1975). Adaptation in Artificial and Neural Systems. Ann Arbor (MI), University of Michigan Press.

K., P. A. The Sinkhorn-Knopp Algorithm: Convergence and Aplications. 26 Richmond Street, Glasgow G1 1XH Scotland, University of Strathclyde, Scotland**:** 1-18.

Karelson, M. (2000). Molecular Descriptors in QSAR/ QSPR. New York, John Wiley & Sons.

Karelson, M. (2000). Molecular Descriptors in QSAR/ QSPR. New York, John Wiley & Sons.

Kniaz, D. (2000). Mod. Drug. Discovery.

Kubinyi, H. (1993). "Parameters in Methods and Principles in Medicinal Chemistry In QSAR Hansch Analysis and related Approaches." Mannhold: 21.

Lawler and Col, Eds. (1985). Algoritmos Geneticos.

Louis, J. C. (2003). Biosilico.

LUKE, S., L. PANAIT, et al. "A Java-based Evolutionary Computation Research System." Evolutionary Computation Journal.

Maltsev, A. I. (1976). Fundamentos del Álgebra Lineal. Moscow, Mir.

Marrero-Ponce, Y.; Castillo-Garit, J. A.; Olazabal, E.; Serrano, H. S.; Morales, A.; Castañedo, N.; Ibarra-Velarde, F.; Huesca-Guillen, A.; Jorge, E.; del Valle, A.; Torrens, F.; Castro, E. A. (2004). J. Comput.-Aided Mol. Design **18**: 615.

Marrero-Ponce, Y. (2003). "Total and Local Quadratic Indices of the Molecular Pseudograph´s Atom Adjacency Matrix: Applications to the Prediction of Physical Properties of Organic Compounds." Molecules **8**: 687-726.

Marrero-Ponce, Y. (2004). "Linear indices of the "molecular pseudograph's atom adjacency matrix": definition, significance-interpretation, and application to QSAR analysis of flavone derivatives as HIV-1 integrase inhibitors." J Chem Inf Comput Sci **44**(6): 2010-2026.

Marrero-Ponce, Y. (2004). "Total and local (atom and atom type) molecular quadratic indices: significance interpretation, comparison to other molecular descriptors, and QSPR/QSAR applications." Bioorg. Med. Chem. **12**: 6351-6369.

Marrero-Ponce, Y., M. A. Cabrera, et al. (2003). "Total and Local Quadratic Indices of the ¨Molecular Pseudograph´s Atom Adjacency Matrix¨. Application to Prediction of Caco-2 Permeability of Drugs. ." Int. J. Mol. Sci. **4**(9): 512-536.

Marrero-Ponce, Y. and V. Romero (2002). TOMOCOMD software TOMOCOMD (TOpological MOlecular COMputer Design). Central University of Las Villas**:** TOMOCOMD software is a preliminary experimental version; in future a professional version can be obtained upon request to Y. Marrero: yovanimp@qf.uclv.edu.cu or ymarrero77@yahoo.es.

Marrero-Ponce, Y. and V. Romero (2002). TOMOCOMD software. TOMOCOMD (TOpological MOlecular COMputer Design) for Windows, version 1.0 is a preliminary experimental version; in future a professional version will be obtained upon request to Y. Marrero: yovanimp@qf.uclv.edu.cu; ymarrero77@yahoo.es. Central University of Las Villas.

Marrero-Ponce, Y. and F. Torrens (2006). "Novel 2D TOMOCOMD-CARDD Descriptors: Atom-based Stochastic and non-Stochastic Bilinear Indices and their QSPR Applications." J. Chem. Inf. Model: Submitted for publication.

MayaChem (2010) Maya Chem Tools.

MERELO, J. J., M. G. ARENAS, et al., Eds. (2000). Evolving Objects. Proc. JCIS 2000 (Joint Conference on Information Sciences).

MERELO, J. J., M. KEIJZER, et al. "EO evolutionary computation framework.". from http://eodev.sourceforge.net.

MICHALEWICZ, Z., Ed. (1992). Genetic Algorithms + Data Structures = Evolution Programs. Berlin, Springer Verlag.

MITCHELL, M., Ed. (1996). An Introduction to Genetic Algorithms. MIT Press. Cambridge (MA).

Noriega, T. (1990). Álgebra. Havana, Cuba, Ed. Revolucionaria.

O'Boyle, N. M. and G. R. Hutchison (2008). "Cinfony." J. Chem. Cent. **2**: 24.

OECD (2004). "The Report from the Expert Group on (Quantitative) Structure–Activity Relationships [(Q)SARs] on the Principles for the Validation of 61 (Q)SARs." ENV/JM/TG(2004)27/REV. In: DEVELOPMENT, O. F. E. C. A.

PARRILL, A. L. (1996). "Drug Discovery Today." **1**: 514-521.

PEDERSEN, J. T. and J. MOULT (1996). "Curr. Opin. Struct. Biol, ." 6.

Project, O. T. O. from http://www.openscience.org.

Randic, M. (1991). J. Math. Chem. **7**: 155.

Ross, K. A. and C. R. B. Wright (1990). Matemáticas discretas. Mexico D.F., Prentice Hall Hispanoamericana.

ROTSTAN, N. and K. MEFFERT (2008). "JGAP: Java Genetic Algorithms Package.". from http://jgap.sourceforge.net/index.html.

Singh, M. G. B. S. M. B. S. (2000). "Pharm. Sci. Technol. Today ": 28.

STEINBECK, C., Y. HAN, KUHN, S., et al. (2003). "The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo and Bioinformatics." Journal of Chemical Information and Computer Sciences **43**: 493-500.

Todeschini, R. and V. Consonni (2000). Handbook of Molecular Descriptors. D-69469 Weinheim, Federal Republic of Germany, WILEY-VCH Verlag GmbH.

Tubert-Brohman, I. "PerlMol Project." from http://ivan.tubert.org/.

van de Waterbeemd, H. (1995). Chemometric Methods in Molecular Design (Methods and Principles in Medicinal Chemistry). New York, John Wiley & Sons.

van de Waterbeemd, H. C., R. E.; Grassy, G.; Kubinyi, H.; Martin, Y. C.; Tute, M., S.; Willett, P. (1998). "Annu. Rep. Med. Chem. ." 33.

WALL, M. (1995). "Overview of Matthew's genetic algorithm library.". from http://lancet.mit.edu/ga,1995.

WEGNER, D. C. J. R. K. (2004). JOELib Tutorial: A Java based cheminformatics/computational chemistry package., Computer Architecture, University of TÃ¼bingen.

WHITLEY, D. L. (1991). "Fundamental principles of deception in genetic search. Foundations of genetic algorithms." San Mateo, CA: Morgan Kaufmann: 221-241.

WHITLEY, D. L. (2002). "Genetic Algorithms and Evolutionary Computing." Van Nostrand's Scientific Encyclopedia.

WILLETT, P. (1995). "Trends Biotechnol." **13**: 512-516.