UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE CIENCIAS EXACTAS

DEPARTAMENTO DE QUÍMICA

Tesis Doctoral

DESARROLLO Y APLICACIÓN DE LA TEORÍA QSAR/QSPR

Andrew Gustavo Mercader

DIRECTOR: Eduardo Alberto Castro

CODIRECTOR: Francisco Marcelo Fernández

2009

El presente trabajo de Tesis se desarrolló en el Instituto de Investigaciones Fisicoquímicas Teóricas y Aplicadas (INIFTA), dependiente del Departamento de Química de la Facultad de Ciencias Exactas, Universidad Nacional de La Plata. Se presenta en consideración de las autoridades de dicha facultad para acceder al Título de Doctor en Ciencias Químicas de la Facultad de Ciencias Exactas.

Agradecimientos

Quisiera agradecer especialmente a todos aquellos que hicieron posible llevar a cabo el trabajo de tesis. A los doctores Eduardo A. Castro y Francisco M. Fernández, por aceptar dirigirme a lo largo de la tesis y por su permanente asistencia e invaluable colaboración. Al doctor Pablo R. Duchowicz por su incomparable colaboración y contribución. Al doctor Julián Echave por su aporte de ideas. A todos los integrantes del INIFTA, ya que todos ellos representan el lugar de trabajo del que me siento orgulloso. A todos los colaboradores científicos que no pertenecen al instituto pero con los que se está en permanente contacto. A la Facultad de Ciencias Exactas de la Universidad Nacional de La Plata, en la cual he sido docente durante gran parte del transcurso de este período en las Cátedras de Fisicoquímica I. Al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), por otorgarme la beca de postgrado que financió el trabajo de tesis.

Dedicado a mi familia, en especial a Juancito.

Índice General

Índice de Figuras

Índice de Tablas

1 Introducción

"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber" (Albert Einstein)

Antiguamente la ciencia química fue catalogada como un campo experimental marcadamente alejado de consideraciones matemáticas. Sin embargo con la llegada de la Mecánica Cuántica los pioneros en ese campo comenzaron a darse cuenta que tenía el potencial de ser una teoría predictiva en química.[1]

En la actualidad se conoce que cuando se pretenden estudiar distintas propiedades de interés presentes en diversos sistemas fisicoquímicos, biológicos o farmacológicos, que dependen tanto de la reactividad química como de la estructura molecular, sin duda alguna deberán abordarse los métodos teóricos derivados de la Mecánica Cuántica con el fin de representar adecuadamente el fenómeno. Sin embargo esto implicaría la necesidad de tomar en consideración todas las interacciones posibles presentes en el sistema físico de partículas.

Ciertamente hay ciertas propiedades físicas que pueden ser calculadas fácilmente usando la química cuántica computacional, ejemplos incluyen momentos dipolares, cargas y entalpías de formación. Sin embargo estos métodos no pueden abordar problemas fisicoquímicos más complejos y ciertamente presentan dificultades en el estudio de sistemas biológicos donde además se debe tener en cuenta la naturaleza del receptor de la molécula en estudio.[2]

A pesar que la potencia computacional a crecido enormemente desde 1929, los químicos cuánticos deben seguir las instrucciones de Paul Dirac " métodos aproximados prácticos deben ser desarrollados que puedan llevar a una explicación de las características principales de un sistema complejo de átomos para no necesitar demasiados cálculos."[1]

Por este motivo se puede afirmar que el desafío principal a superar en los años venideros será la descripción de sistemas constituidos por miles de átomos interactuando entre sí; esto es así porque los cálculos mecano cuánticos actuales sólo pueden resolverse con buena aproximación usando un nivel de teoría adecuado, siempre y cuando involucren moléculas de unos pocos átomos y se hallen libres de interacciones

intermoleculares. Es razonable suponer que si bien pueden postularse métodos con severas aproximaciones para resolver el problema de partículas interrelacionadas entre sí, su naturaleza incierta hará que no siempre se haga posible explicar satisfactoriamente y de manera comprensible la calidad de los resultados finales obtenidos.

De lo anterior se desprende una pregunta, cómo podemos calcular las propiedades y actividades de sistemas complejos? La respuesta es que en lugar de calcular podemos hacer un modelo basado en datos experimentales para predecir directamente la propiedad/actividad.

1.1 El uso de modelos en ciencias exactas

Es sabido que la mayoría de las aplicaciones de análisis de datos involucra intentos de ajustar un modelo, usualmente del tipo cuantitativo, a un conjunto de medidas experimentales u observaciones. Las razones para ajustar tales modelos son variadas. Por ejemplo, el modelo puede ser puramente empírico y ser requerido a efectos de hacer predicciones para realizar nuevos experimentos. Por otra parte, el modelo puede estar basado en alguna teoría o ley, y una evaluación del ajuste de los datos al modelo puede ser usada para ganar una perspectiva del proceso subyacente en las observaciones efectuadas.

En algunos casos la habilidad para ajustar satisfactoriamente un modelo a un conjunto de datos puede ofrecer elementos útiles para formular nuevas hipótesis. El tipo de modelo que puede ser ajustado a algún conjunto de datos depende no solamente de la naturaleza de los datos sino que también del uso que se hará del modelo. En muchas aplicaciones un modelo se supone que será utilizado predictivamente, pero las predicciones no tienen que ser necesariamente cuantitativas.[3, 4]

Se podría decir que el origen de toda teoría se basa en la observación, recolección de datos experimentales y modelado que deriva en ecuaciones matemáticas a las cuales luego se les busca un significado físico.

En algunas circunstancias puede suceder que el análisis de datos no ajuste un modelo. El sencillo procedimiento de graficar los valores de dos variables puede no constituir de modo alguno algo asociado al modelado, a menos que se conozca de antemano que las variables están vinculadas por medio de alguna ley. La producción de un gráfico bidimensional se puede pensar como el ajuste de un modelo que está

sencillamente dictado por las variables. Esto puede parecer un concepto extraño pero es una forma útil de visualizar qué está sucediendo cuando las técnicas multivariadas se aplican para desplegar los datos. Los gráficos resultantes se pueden pensar como modelos que se han ajustado a los datos y como resultado de ello puede brindar alguna perspectiva acerca de la información que el modelo, y por ende los datos, contiene.[5, 6]

1.2 *Breve reseña histórica*

El hecho de que distintos compuestos químicos tienen diferentes efectos biológicos se conoce desde hace muchísimo tiempo.

Hace más de cien años Crum-Brown y Fraser expresaron la idea que la actividad fisiológica de una sustancia era función de su composición química.[7] Algunas décadas más tarde, en 1893, Richet mostró que la citotoxicidad de un diverso grupo de moléculas orgánicas simples estaban inversamente relacionadas con su solubilidad en agua.[8] A fines del siglo XIX Meyer and Overton sugirieron en forma independiente que la actividad narcótica de un grupo de compuestos orgánicos estaba directamente relacionada con su constante de partición entre agua y aceite de oliva.[9, 10] En 1939 Ferguson introdujo una generalización a la correlación de la actividad narcótica con la saturación relativa de compuestos volátiles en el medio de administración.[11] El trabajo extenso de Albert, Bell y Roblin estableció la importancia de la ionización de las bases y ácidos débiles en la actividad bacteriostática.[12-14] Mientras tanto el trabajo de Hammet en el campo de la física orgánica dio comienzo a la delineación del efecto del sustituyente en reacciones orgánicas.[15, 16] Taft encontró una forma de separar los efectos polares, estéricos y de resonancia e introdujo el primer parámetro estérico, *Es*.[17] Las contribuciones de Hammett y Taft juntos dejaron las bases mecánicas para el desarrollo del paradigma QSAR por Hansh y Fujita en 1964;[18] desde entonces los avances en esta materia no han cesado y los aportes significativas se suceden continuamente.

El estudio de las Relaciones Cuantitativas Estructura-Propiedad (QSPR) y las Relaciones Cuantitativas Estructura-Actividad (QSAR), continúa atrayendo la atención de forma considerable en la literatura química corriente, y aún en terrenos tales como la Física, la Farmacología y Química Analítica, incluyendo particularmente a la Fisicoquímica, la Química Medicinal, la Quimiometría, la Teoría de Grafos Químicos,

la Química Matemática, la Química Computacional, el Modelado Matemático, la Manipulación de Estructuras Asistida por Computadora, la Inteligencia Artificial, y otras más.

1.3 Importancia de las teorías QSPR-QSAR

Los estudios QSPR-QSAR combinan métodos de la Mecánica Estadística con la Química Computacional, con el objeto de establecer modelos matemáticos que cuantifiquen las relaciones estructura-actividad (SAR), estas últimas son simplemente las observaciones que un cierto cambio en la estructura química de un determinado compuesto posee un cierto efecto sobre la actividad biológica.[19, 20] La importancia vital de las aplicaciones de esta teoría aparece cuando se dispone del conocimiento de la estructura química, lo que permite predecir el valor de la propiedad experimental de una determinada sustancia si ésta permanece desconocida por resultar difícil de analizar, ya sea por tratarse de sustancias inestables o tóxicas o por ser poco accesibles económicamente o demandar mucho tiempo de síntesis.

Un ejemplo de la importancia que adquieren este tipo de estudios es el siguiente: el potencial de la química orgánica para la producción de nuevos compuestos es realmente enorme, ya sea que ellos estén destinados a usos farmacéuticos o agroquímicos, sean para generar fragancias, saborizantes o alimentos. En 1994, el Chemical Abstracts presentó más de 13 millones de compuestos, pero sólo una pequeña proporción de ellos se pudo llegar a sintetizar.

¿Cómo seleccionar aquellos compuestos que vale la pena sintetizar? Si la cuestión se resolviera por el mero procedimiento de síntesis y testeo sin otra guía que la posterior prueba y error, ello constituiría una metodología puramente empírica, extremadamente laboriosa, costosa y nada científica.[21, 22]

La teoría QSPR-QSAR también puede emplearse para describir a la propiedad en términos estructurales sugiriendo así paralelismos, para el descubrimiento, diseño y optimización molecular de nuevas drogas[23, 24] y hasta para suministrar alguna información en cuanto a los mecanismos de las reacciones químicas. Una de las ventajas de poseer mejores descripciones de la estructura molecular es que resulta posible transferir información de una serie de moléculas a otra serie distinta.[25-27]

1.4 *Objetivo y fundamentos de las teorías QSPR-QSAR*

El objetivo de las teorías QSAR y QSPR es predecir propiedades o actividades biológicas usando información que proviene de la estructura molecular. Esto permite predecir propiedades y actividades que aún no han sido medidas experimentalmente, incluso para sustancias que no existen debido a que todavía no han sido sintetizadas.[3]

El fundamento de las teorías se basa en el hecho que la propiedad o actividad de una sustancia depende únicamente de su estructura molecular. Sin embargo cómo puede verse en la Figura 1.4.1 la flecha que representa el vínculo directo está rota porque la relación permanece desconocida.

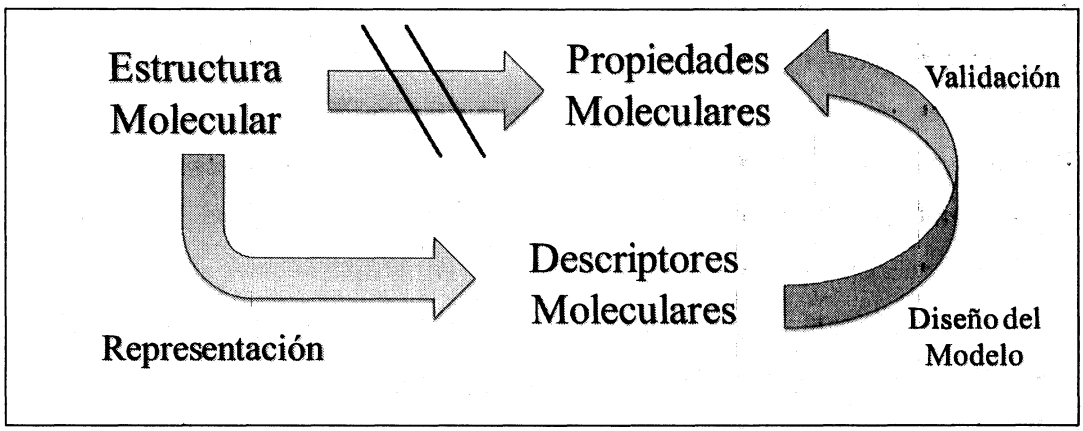

Figura 1.4.1 Fundamentos de las teorías QSAR/QSPR

Con objeto de establecer la relación QSPR-QSAR es un requisito indispensable calibrar el modelo con un conjunto de moléculas para las cuales se conocen los valores experimentales de la propiedad, denominado conjunto molecular de calibración. Esto nos lleva a entender que se trata de una teoría inductiva y semiempírica. La función matemática lineal o no-lineal con la que se construye el modelo se escoge arbitrariamente y el criterio práctico es apelar a aquella que brinde los mejores resultados.

La conexión estructura-propiedad se puede establecer con varios métodos estadísticos que se consideran herramientas útiles como técnicas de reducción de datos, tales como: el Análisis de Regresión Multivariable (MRA),[25] el Análisis de Componentes Principales (PCA),[26] el Método de Cuadrados Mínimos Parciales (PLS),[27] y las Redes Neuronales Artificiales (ANN).[27] Inclusive, varios

procedimientos se han empleado de manera combinada, como por ejemplo una combinación de las ANN y PCA.[28] Asimismo, no es necesario contar con un método estadístico para establecer el modelo, como en el caso del Método de Ordenamiento Parcial (POR). El análisis de regresión multivariable es el método estadístico multidisciplinario que más ampliamente se ha usado desde hace muchos años en el campo de la teoría QSPR-QSAR. Los restantes métodos constituyen propuestas de mejoras respecto de este último y han sido introducidos en los últimos 30 años. Dependiendo de la técnica que se utilice en el análisis hará que los modelos QSPR-QSAR difieran en su complejidad, precisión, estabilidad, capacidad predictiva y de interpretación. Cabe mencionar que la aproximación QSPR-QSAR ha evolucionado a lo largo de los años desde un modelo de regresión simple con pocas variables[18] hasta transformarse en una herramienta importante aplicable a un amplio rango de problemas químicos, biológicos, medicinales y farmacológicos.[28-30]

En el campo de QSPR-QSAR la estructura molecular se define como la disposición tridimensional de los átomos en una molécula. Para extraer la información de la estructura se usan los denominados descriptores moleculares,[31] estos son, variables numéricas que reflejan distintos aspectos de la estructura molecular y que pueden tratarse tanto de cantidades teóricas calculadas así como de cantidades experimentales. Ellos son construidos usando la Teoría de Grafos, aproximaciones de la Química Cuántica, de la Teoría de la Información o mediante Aproximaciones Geométricas.[32-35]

Hoy día se cuenta con miles de descriptores disponibles[32] y uno debe enfrentarse con el problema de la adecuada selección de las mejores variables estructurales a partir de un conjunto enorme[33], teniendo que decidir con cuáles y cuántas se diseña el modelo[34, 35]. Se desea que la relación cuantitativa posea pocos descriptores ya que de esta manera resultará fácil establecer un paralelismo simple entre la propiedad investigada y la estructura, observando las regularidades generales de los datos experimentales en unos pocos parámetros representativos que permitan entenderla. Si el modelo contiene muchos descriptores entonces puede conducir a confusiones y controversias al tratar de analizarlo. Es una regla aceptada en la práctica que al establecer un modelo matemático debe comenzarse desde lo más simple, y solo si es necesario, proceder a un nivel más complejo en su formulación matemática. No tiene sentido armar un modelo complicado si otro que sea mucho más simple logra resultados semejantes. Si el modelo simple falla en las predicciones, entonces recién ahí se puede

diseñar uno más avanzado introduciendo funciones no-lineales en la ecuación y recurriendo a varios parámetros en vez de unos pocos, aunque sin duda alguna esto oscurecerá el rol de las variables que dan origen a las predicciones del modelo.

Una vez seleccionado el conjunto de moléculas que exhibe cierta propiedad a estudiar, la función matemática y los descriptores óptimos, se pueden establecer las relaciones QSPR-QSAR calibrando el modelo. El siguiente paso en el estudio, y que es de fundamental importancia, consiste en analizar el poder predictivo del modelo encontrado llevando a cabo una validación del mismo. De esta forma, el modelo no solo se limitará a reproducir los datos de la propiedad de las moléculas usadas en la calibración, sino que también efectuará predicciones para un conjunto de moléculas estructuralmente relacionadas pero distintas a las empleadas durante la calibración, que representan el conjunto de validación. Para esto al inicio del estudio se deja apartado un grupo de moléculas que se denomina como conjunto de validación (o en inglés Test Set) que no participan en la calibración del modelo. Asimismo se emplean métodos de validación cruzada, siendo estos muy útiles para casos donde haya disponibles pocos datos experimentales. Estos métodos se basan en la remoción de moléculas de a una o varias a la vez para luego re-calibrar el modelo y predecir la propiedad de la molécula/s que fue removida, comparando luego el valor predicho con el experimental. Estos métodos son conocidos como validaciones cruzadas de *Leave-One-Out (loo)* y *Leave-More-Out (l-n%-o)* [36].

1.5 *Estado actual de las teorías QSAR/QSPR*

En la actualidad los autores del campo QSPR-QSAR intentan realizar diversas mejoras en la teoría, tales como

- proponer descriptores moleculares noveles (aquellos que engloben toda la información estructural de la molécula en una única variable y que a su vez sea aplicable a cualquier propiedad-actividad bajo estudio[37]) y más poderosos
- desarrollar nuevas estrategias para obtener los modelos matemáticos QSPR-QSAR.[38-40]
- establecer el numero optimo de descriptores en un modelo
- desarrollar nuevos métodos de validación

- aplicar las teorías a conjuntos de datos que no hayan sido modelados o que no se ha encontrado un modelo predictivo

El resultado del primer punto lleva a la ya mencionada sobreabundancia de descriptores. El segundo punto conduce a un continuo mejoramiento en la metodología de modelado y selección de descriptores.

Además cabe resaltar que las teorías QSPR-QSAR se encuentran desarrolladas y en un estado avanzado, de manera que sus resultados se vuelven cada vez más dependientes de los recursos computacionales disponibles al presente. Por lo que también se busca realizar optimizaciones computacionales de los métodos ya existentes.

1.6 Objetivos del trabajo de Tesis

El presente trabajo tuvo como objetivos el estudio, desarrollo y aplicación de las teorías QSPR-QSAR. Para esto se abordaron diversos temas, tal como se podrá ver en los capítulos venideros, estudiando los diferentes puntos presentados en el inciso anterior.

El capítulo 2 describe la forma óptima de representar y optimizar las moléculas para este tipo de estudios.

El capítulo 3 exhibe un resumen de los descriptores moleculares disponibles en la actualidad como así los descriptores nuevos estudiados. Cabe mencionar que no se hizo hincapié en el estudio de este tema particular debido a la inmensa cantidad de descriptores disponibles que abarcan la mayoría de las propiedades moleculares y a que en la opinión del autor no es trascendental la obtención de descriptores noveles.

El capítulo 4 muestra un resumen de los métodos de búsqueda para el diseño de modelos y los nuevos métodos desarrollados. Además se mostrará un método novedoso para determinar el número óptimo de descriptores a incluir en un modelo.

El capítulo 5 presenta los métodos de validación existentes, usados en la aplicación de la teoría y una propuesta novedosa.

El capítulo 6 expone las aplicaciones realizadas durante el trabajo de Tesis de las teorías a diversos problemas de interés.

El capítulo 7 contiene los desarrollos computacionales que permitieron disminuir enormemente los tiempos de cálculo.

2 Representación de la Estructura Molecular

"Lo que sabemos es una gota de agua; lo que ignoramos es el océano"
(Isaac Newton)

2.1 Introducción

A diferencia de otras disciplinas científicas que solo usan textos y números para transferir información, la Química tiene un desafío adicional, representar a las moléculas. Las especies moleculares consisten de átomos y enlaces que los sujetan. Adicionalmente, estos compuestos pueden convertirse en otros por reacciones químicas. Por lo tanto la información relacionada a la química no solo abarca texto y datos pero también que caracterizar compuestos químicos con estas propiedades especiales y sus reacciones.

En un principio solo nombres fueron usados para caracterizar los compuestos. Muy pronto símbolos fueron usados para acortar nombres largos. La clasificación sistemática de compuestos de acuerdo con su naturaleza como así técnicas experimentales llevaron a mejorar el conocimiento de la estructura molecular, esto eventualmente llevó a asignar a los compuestos los conocidos diagramas de estructuras y luego su arreglo tridimensional.

La representación bidimensional en diagramas puede considerarse un lenguaje nativo universal para los químicos. Estos diagramas son modelos y están diseñados para mejorar la percepción de las moléculas. En estos modelos los átomos se representan por el símbolo de su elemento y los enlaces por líneas. Sin embargo un diagrama es una representación incompleta y altamente simplificada de una molécula; solo explica la topología (que átomos están conectados por qué tipo de enlaces) y no el arreglo tridimensional (topografía) de los átomos en la molécula. Como es conocido la estructura tridimensional requiere información adicional como la posición de los átomos en el espacio o los ángulos y distancias entre átomos. Aún información más compleja es necesaria para poder conocer propiedades como el potencial electroestático, para esto se necesita mapear una superficie tridimensional. En la Figura 2.1.1 se pude observa un esquema de la clasificación según la complejidad de la representación.[41]

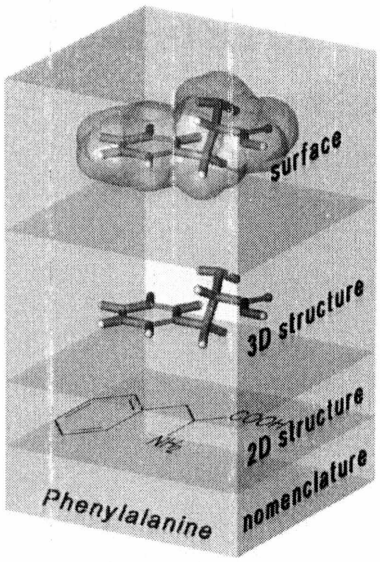

Figura 2.1.1 Esquema de la jerarquización según la complejidad de la representación[41]

Para poder hacer uso del poder de las computadoras uno de los desafíos es poder traducir la estructura de un compuesto en un código interpretable por programas de aplicación. Existen diversos caminos para encarar esta tarea, los mismos han ido evolucionando a lo largo del tiempo, por problemas de espacio solo se trataran los relacionados a este trabajo, es decir a las representaciones necesarias para los cálculos QSAR/QSPR.

2.2 Teoría de Grafos

Uno de los caminos usa la teoría de grafos, la cual es usada en Matemáticas para describir una gran variedad de problemas y situaciones,[42] como por ejemplo el problema de los puentes de Königsberg resuelto por Euler en 1736, quien a partir de este problema introdujo la teoría de grafos.[43] La transferencia de modelos y abstracciones de otras ciencias a la teoría de grafos permite hacer un procesamiento matemático simple. La analogía entre un diagrama químico y un grafo topológico es la base del desarrollo de algoritmos teóricos provenientes de la teoría de grafos para procesar información estructural química.[44-46]

En términos matemáticos, los diagramas químicos pueden ser considerados como grafos comunes. Los grafos consisten en nodos (vértices), que hacen las veces de

átomos, y líneas o aristas que serían los enlaces. Este tipo de grafos se denomina grafo topológico ya que solo muestra la conexión entre átomos y el tipo de enlace.

En esta teoría un grafo no contiene ninguna información geométrica como muestra la Figura 2.2.1.

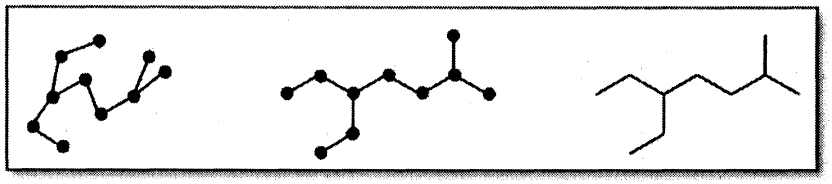

Figura 2.2.1 Representaciones diferentes de un diagrama idéntico

Un grafo pesado tiene números o símbolos en los nodos. Dos nodos pueden tener varias aristas que los conecten (Figura 2.2.2).

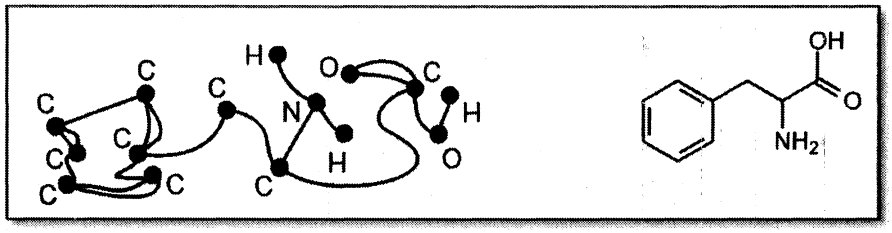

Figura 2.2.2 La Fenilamina puede representarse como un grafo pesado por distintos átomos

2.3 Representaciones mediante Matrices

Un grafo puede ser representado por una Matriz. La mayor ventaja es que los cálculos se pueden llevar a cabo por las muy conocidas operaciones matriciales. La mejor forma de visualizar estas representaciones es mediante un ejemplo; en la Tabla 2.3.1 se puede ver la matriz de adyacencia o conectividad, para el acetaldehído (Figura 2.3.1). La intersección de una columna con una fila recibe un número 1 si los átomos correspondientes tienen un enlace. Se puede observar además cómo las matrices pueden ser simplificadas obviando información redundante.

Tabla 2.3.1 Matriz de adyacencia para el acetaldehído (Fígura 2.3.1). a) Matriz redundante b)simplificada quitando los ceros c)reduciéndola al triangulo superior d) omitiendo los hidrógenos

a	1	2	3	4	5	6	7
1	0	1	0	1	1	1	0
2	1	0	1	0	0	0	1
3	0	1	0	0	0	0	0
4	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0
6	1	0	0	0	0	0	0
7	0	1	0	0	0	0	0

b	1	2	3	4	5	6	7
1		1		1	1	1	
2	1		1				1
3		1					
4	1						
5	1						
6	1						
7		1					

c	1	2	3	4	5	6	7
1		1		1	1	1	
2			1				1
3							
4							
5							
6							
7							

d	1	2	3
1		1	
2			1
3			

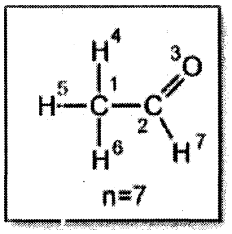

Figura 2.3.1 Estructura del acetaldehído

La representación de la estructura es un paso esencial en cualquier estudio QSPR-QSAR, ya que existen diversas matrices para representar distintos aspectos de la molécula en la Tabla 2.3.2 se encuentra un resumen de las mismas mostrando sus principales ventajas y desventajas.

Tabla 2.3.2 Resumen y evaluación de la representación matricial de estructuras químicas

Ventajas	Desventajas
Matriz General: aspectos generales de la representación matricial	
• **El diagrama molecular se codifica completamente(cada átomo y enlace es representado)** • **Se puede usar álgebra matricial**	• El número de entradas en la matriz crece con el cuadrado del número de átomos • No se incluye estereoquímica
Matriz de adyacencia: las columnas y filas representan los átomos(nodos)	
• **Describe la conexión de los átomos** • **Solo contiene ceros y unos (bits)**	• No se representan el tipo ni el orden de los enlaces • No se representan el número de electrones libres

Matriz de distancia: la distancia se expresa como geométrica (Å) o topológica (número de enlaces)	
• **Describe la distancia geométrica**	• No se representan el tipo ni el orden de los enlaces • No se representan el número de electrones libres • No se puede representar por bits
Matriz de incidencia: las columnas representan los átomos (nodos) y las filas los enlaces (vértices)	
• **Describe las conexiones y enlaces** • **Solo contiene ceros y unos (bits)**	• No se representan el tipo ni el orden de los enlaces • No se representan el número de electrones libres
Matriz de enlaces: las columnas y filas representan los átomos (nodos). Si existe enlace múltiple se representa con el número correspondiente, Ej.: doble enlace 2.	
• **Describe las conexiones y ordenes de enlaces**	• No se representan el número de electrones libres • No se puede representar por bits
Matriz de enlaces-electrón: Puede considerarse como una extensión de la Matriz de enlaces. En esencia enumera todos los electrones de valencia en la molécula. Estas fueron introducidas por Dugundji-Ugi[47]	
• **Describe las conexiones y ordenes de enlaces y electrones de valencia**	• No se puede representar por bits

2.4 Tablas de conexión

Para solucionar el problema del aumento cuadrático de entradas con el número de átomos de las Matrices, se utilizan las denominadas tablas de conexión (CT). Estas constan de un listado de los átomos y un listado de los enlaces que los conectan, su número de entradas aumenta de forma lineal con el aumento del número de átomos de la estructura molecular a ser representada.

Las tablas de conexión han sido la manera predominante de representar estructuras químicas en sistemas informáticos desde los comienzos de la década del 80. Estas son una alternativa a la representación de un grafo molecular, pudiendo aplicárseles igualmente bien la teoría de grafos.

En la Tabla 2.4.1 se pude observar una tabla de conexión para el acetaldehído, si bien la misma posee información redundante podría luego ser simplificada.

Tabla 2.4.1 Tabla de conexión del Acetaldehído

Atomo número	Elemento	Conectado con:	Orden del enlace	Conectado con:	Orden del enlace	Conectado con:	Orden del enlace	Conectado con:	Orden del enlace
1	C	2	1	4	1	5	1	6	1
2	C	1	1	3	2	7	1		
3	O	2	2						
4	H	1	1						
5	H	1	1						
6	H	1	1						
7	H	2	1						

2.5 Formato de intercambio de estructuras

Existen un gran número de programas informáticos para el manejo y procesamiento de estructuras químicas. Algunos simplemente son usados para dibujar las mismas en documentos de texto y otros para obtener información adicional de la estructura mediante distintos cálculos.

Muchas organizaciones y compañías de software han desarrollado sus propios formatos de tablas de conexión y pocas se han tomado el tiempo de incluir en sus programas la posibilidad de importar y exportar moléculas con distintos formatos.

Por este motivo sería muy útil que existiese un formato universal de representación de moléculas, sin embargo todavía no existe tal y uno debe en muchos casos extraer a mano de las tablas de conexión la información necesaria para otro tipo de formato. Afortunadamente se están desarrollando programas de traducción de formatos, como el programa de distribución gratuita OpenBabel[48]. El mismo permite traducir entre más de 100 formatos de estructuras distintos, los cuales fueron enumerados en la Tabla 2.5.1 con el fin de mostrar la inmensa cantidad y variedad de formatos disponibles.

Tabla 2.5.1 Formatos electrónicos de estructuras químicas que permite traducir el OpenBabel[48]

acr -- Carine ASCI Crystal	feat -- Feature format	mpc -- MOPAC Cartesian format
alc -- Alchemy format	fh -- Fenske-Hall Z-Matrix format (Write-only)	mpd -- Sybyl descriptor format (Write-only)
arc -- Accelrys/MSI Biosym/Insight II CAR format	fix -- SMILES FIX format (Write-only)	mpqc -- MPQC output format (Read-only)
bgf -- MSI BGF format	fpt -- Fingerprint format (Write-only)	mpqcin -- MPQC simplified input format (Write-only)
box -- Dock 3.5 Box format	fract -- Free Form Fractional format	nw -- NWChem input format (Write-only)
bs -- Ball and Stick format	fs -- Open Babel FastSearching database	nwo -- NWChem output format (Read-only)
c3d1 -- Chem3D Cartesian 1 format	fsa -- FASTA format (Write-only)	pc -- PubChem format (Read-only)
c3d2 -- Chem3D Cartesian 2 format	g03 -- Gaussian 98/03 Output (Read-only)	pcm -- PCModel format
caccrt -- Cacao Cartesian format	g98 -- Gaussian 98/03 Output (Read-only)	pdb -- Protein Data Bank format
cache -- CAChe MolStruct format (Write-only)	gam -- GAMESS Output (Read-only)	pov -- POV-Ray input format (Write-only)
cacint -- Cacao Internal format (Write-only)	gamin -- GAMESS Input (Write-only)	pqs -- Parallel Quantum Solutions format
can -- Canonical SMILES format	gamout -- GAMESS Output (Read-only)	prep -- Amber Prep format (Read-only)
car -- Accelrys/MSI Biosym/Insight II CAR format (Read-only)	gau -- Gaussian 98/03 Cartesian Input (Write-only)	qcin -- Q-Chem input format (Write-only)
ccc -- CCC format (Read-only)	gjc -- Gaussian 98/03 Cartesian Input (Write-only)	qcout -- Q-Chem output format (Read-only)
cdx -- ChemDraw binary format (Read-only)	gjf -- Gaussian 98/03 Cartesian Input (Write-only)	report -- Open Babel report format (Write-only)
cdxml -- ChemDraw CDXML format	gpr -- Ghemical format	res -- ShelX format (Read-only)
cht -- Chemtool format (Write-only)	gr96 -- GROMOS96 format (Write-only)	rxn -- MDL RXN format
cif -- Crystallographic Information File	hin -- HyperChem HIN format	sd -- MDL MOL format
cml -- Chemical Markup Language	inchi -- IUPAC InChI (Write-only)	sdf -- MDL MOL format
cmlr -- CML Reaction format	inp -- GAMESS Input (Write-only)	smi -- SMILES format
com -- Gaussian 98/03 Cartesian Input (Write-only)	ins -- ShelX format (Read-only)	sy2 -- Sybyl Mol2 format
copy -- Copies raw text (Write-only)	jin -- Jaguar input format (Write-only)	tdd -- Thermo format
crk2d -- Chemical Resource Kit 2D diagram format	jout -- Jaguar output format (Read-only)	test -- Test format (Write-only)
crk3d -- Chemical Resource Kit 3D format	mdl -- MDL MOL format	therm -- Thermo format
csr -- Accelrys/MSI Quanta CSR format (Write-only)	mmd -- MacroModel format	tmol -- TurboMole Coordinate format
cssr -- CSD CSSR format (Write-only)	mmod -- MacroModel format	txyz -- Tinker MM2 format (Write-only)
ct -- ChemDraw Connection Table format	mol -- MDL MOL format	unixyz -- UniChem XYZ format
dmol -- DMol3 coordinates format	mol2 -- Sybyl Mol2 format	vmol -- ViewMol format
ent -- Protein Data Bank format	molreport -- Open Babel molecule report (Write-only)	xed -- XED format (Write-only)
fa -- FASTA format (Write-only)	moo -- MOPAC Output format (Read-only)	xml -- General XML format (Read-only)
fasta -- FASTA format (Write-only)	mop -- MOPAC Cartesian format	xyz -- XYZ cartesian coordinates format
fch -- Gaussian formatted checkpoint file format (Read-only)	mopcrt -- MOPAC Cartesian format	yob -- YASARA.org YOB format
fchk -- Gaussian formatted checkpoint file format (Read-only)	mopin -- MOPAC Internal	zin -- ZINDO input format (Write-only)
fck -- Gaussian formatted checkpoint file format (Read-only)	mopout -- MOPAC Output format (Read-only)	

2.5.1 Software empleado para la representación de estructuras

En este trabajo de tesis se utilizó el programa Hyperchem[49] para la representación de estructuras químicas y su posterior optimización.

El motivo principal fue su simplicidad de uso y el hecho de que tiene un formato de salida compatible con el software de cálculo de descriptores. Asimismo la calidad de las optimizaciones de geometría alcanzables permite el cálculo de los descriptores tridimensionales sin inconvenientes.

Para el cálculo de estos últimos las estructuras deben ser llevadas a una posición estable; para esto normalmente se pre optimizan con un método de *Molecular Mechanics Force Field* (MM+) y luego se refina la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere, ambos métodos están incluidos en el paquete de Hyperchem.

Los archivos usados por Hyperchem tienen extensión ".hin" y su tabla de conexión para el acetaldehído (Figura 2.3.1) tiene la estructura mostrada en la Tabla 2.5.2. Comprender la misma tiene mucha importancia para poder llevar a cabo la creación de descriptores constitucionales.

Tabla 2.5.2 Estructura de la tabla de conexión usada por Hyperchem[49]

Número de átomo	Elemento	Tipo	Carga del átomo	Coordenadas cartesianas			Nº de enlaces	Átomos conectados y tipo de enlace							
				x	y	z									
atom 1	C	C4	-0.1959	-0.2601	-1.0482	2.080E-06	4	2	s	4	s	5	s	6	s
atom 2	C	CO	0.2811	-0.2882	0.4522	-3.690E-07	3	1	s	3	d	7	s		
atom 3	O	O1	-0.3182	-1.3141	1.0939	-2.580E-06	1	2	d						
atom 4	H	H	0.0526	0.7675	-1.4317	-2.720E-06	1	1	s						
atom 5	H	H	0.0695	-0.7677	-1.4579	8.836E-01	1	1	s						
atom 6	H	H	0.0695	-0.7677	-1.4579	-8.836E-01	1	1	s						
atom 7	H	H	0.0415	0.6821	0.9747	2.130E-07	1	2	s						

3 Descriptores Moleculares

"Los problemas significativos que nos encontramos no pueden ser resueltos con el mismo nivel de pensamiento que tenemos al formularlos" (Albert Einstein)

3.1 Introducción

Un paso imprescindible en cualquier estudio QSAR/QSPR es el cálculo de descriptores moleculares, estos son el resultado final de un procedimiento lógico y matemático que transforma información química codificada dentro de una representación simbólica de una molécula en un número útil.

Debe prestarse atención al término "útil" ya que tiene dos significados: por un lado que el número pude dar una mayor comprensión de las propiedades moleculares y por el otro, que puede formar parte de un modelo para la predicción de una propiedad de interés de otras moléculas.

El campo de los descriptores moleculares es altamente interdisciplinario e involucra una gran cantidad de teorías diferentes. Esto hace que como se mencionara en la Introducción existan miles de descriptores disponibles en la literatura por lo que sería imposible entrar en detalle sobre el cálculo y características de cada uno de ellos, esto llevaría un espacio similar al requerido por un Handbook[32]. Por estos motivos, el objetivo de este capítulo es presentar las distintas clases de descriptores existentes, proporcionando una visión general de sus características que incluya una breve descripción de la naturaleza de sus cálculos, con el propósito de dejar un entendimiento sólido sobre el tema en forma sintética.

3.2 Clasificación de descriptores moleculares

A grandes rasgos los descriptores moleculares pueden clasificarse según representen a la molécula como un todo, descriptores globales o empleando una aproximación de subestructura o atómica, es decir, por medio de descriptores atómicos que describan los fragmentos presentes en una estructura. Ejemplos de descriptores

moleculares globales son el volumen, el área molecular, la energía electrónica total, etc., y descriptores atómicos son el número de átomos y enlaces.

Asimismo los descriptores pueden clasificarse según la representación de la molécula, en este caso se dividen en descriptores de dimensión: cero (0D), uno (1D), dos (2D) y tres (3D).

3.3 Descriptores 0D

Los descriptores 0D describen la constitución de la molécula y son independientes de la conectividad o conformación molecular; los más sencillos son el número de átomos y enlaces de un determinado tipo, peso molecular, peso atómico promedio, suma de propiedades atómicas como puede ser volúmenes de Van der Waals, etc. Con esta familia de descriptores no se puede diferenciar a los isómeros. La utilidad de estos descriptores se aprecia, por ejemplo, en que el número de carbonos terciarios refleja en cierta medida la forma molecular. El número de átomos H indicaría su capacidad de participar en puentes de hidrógeno, cuyo efecto sobre varias propiedades es notorio.

3.4 Descriptores 1D

Estos son descriptores que involucran fracciones de la molécula en un subconjunto de átomos. Dentro de los descriptores 1D están el número de grupos funcionales, estos son descriptores basados en la cuenta de grupos funcionales químicos. Ejemplos de estos son: número total de átomos de carbono primario (sp3), número de cianatos, número de nitrilos, etc.

Otro grupo de descriptores 1D son los llamados "fragmentos centrados en el átomo", estos son descriptores que se basan en la cuenta de distintos fragmentos de la molécula.[50] Ejemplos de estos son: hidrógenos enlazados a un heteroátomo, Hidrógenos enlazados a un carbono alfa, Flúor enlazado a un Carbono primario (sp3).

3.5 *Descriptores 2D o Topológicos*

Descriptores moleculares obtenidos de la teoría de grafos, independientes de la conformación de la molécula.

3.5.1 Descriptores de Autocorrelacion-2D

Diferentes variables estructurales introducidas por Broto, Moreau y Geary [51, 52] corresponden a autocorrelaciones bidimensionales entre pares de átomos en la molécula y fueron definidas para reflejar la contribución de una propiedad atómica a la propiedad molecular en estudio. La propiedad con la que se ajusta o pesa al descriptor considera los tipos de átomos presentes a través de la electronegatividad (*e*), masa atómica (*m*), polarizabilidad atómica (*p*) o volumen de Van der Waals (*v*), y de esta manera se puede seleccionar a partir de los átomos que componen la molécula, aquellos que dan más peso a la variable. Estos descriptores pueden calcularse fácilmente sumando los productos de los pesos atómicos de los átomos terminales de todos los caminos del largo requerido.

3.5.2 Descriptores BCUT

Estos descriptores son autovalores de una matriz conectividad modificada, la matrix de Burden (**B**)[53, 54]. Esta matriz es un grafo molecular despojado de hidrógenos que se define de la siguiente manera:

- Los elementos diagonales son los números atómicos de los elementos (Z_i)
- Elementos fuera de la diagonal (B_{ij}), representando átomos enlazados *i* y *j* son iguales a $\pi^*.10^{-1}$, donde π^* es el orden de enlace convencional (i.e.: 1, 2, 3, 1.5 para simple, doble, triple y aromático respectivamente)
- Elementos fuera de la diagonal correspondientes a enlaces terminales se incrementan en 0.01
- Todo el resto de los elementos de la matriz se fija en 0.001

La secuencia ordenada de los **n** autovalores más pequeños de **B** se propuso como descriptores moleculares basándose en la suposición que el autovalor más bajo contiene

contribuciones de todos los átomos y por lo tanto refleja la topología de la molécula. Los descriptores BCUT son una extensión de los atuovalores de Burden y consideran tres clases de matrices cuyos elementos diagonales corresponden a valores relacionados con la carga atómica, polarizabilidad y habilidades de unirse a un átomo de hidrógeno.

Se ha demostrado que estos descriptores reflejan aspectos relevantes de la estructura molecular, y se los utiliza en estudios de similitud/diversidad de grandes bases de datos.[55]

3.5.3 Índices de Carga Topológica de Galvez

Estos índices describen la transferencia de carga entre pares de átomos y por lo tanto a la transferencia de carga global en la molécula. Para obtener definiciones de los índices primero definimos la matriz **M** como:

$$\mathbf{M} = \mathbf{A}.\mathbf{D}^{-2} \qquad (3.5.1)$$

donde **A** es la matriz de adyacencia y $\mathbf{D}^{-2}$ es el reciproco al cuadrado de la matriz distancia (ver sección 2.3), notar que los elementos diagonales de la matriz distancia permanecen inalterados. **M** es la matriz de Galvez y es una matriz cuadrada asimétrica de a x a donde a es el número de átomos en la molécula. **M** da lugar a la matriz asimétrica de transferencia de carga, **CT** definida cómo:

$$CT_{ij} = \begin{cases} \delta_i & si\ i = j \\ m_{ij} - m_{ji} & si\ i \neq j \end{cases} \qquad (3.5.2)$$

donde m_{ij} son los elementos de **M** y δ_i es el grado del vértice del átomo i. Las entradas diagonales de **CT** representan la valencia topológica de los átomos y las entradas fuera de la diagonal CT_{ij} indican la cantidad de carga transferida desde el átomo i al j.

Si son considerados los heteroátomos las entradas diagonales de **A** pueden sustituirse por la electronegatividad de Pauling o el grado de valencia del vértice que para un átomo i está definido cómo:

$$\delta_i^v = \sigma_i + \pi_i + n_i - h_i \qquad (3.5.3)$$

donde σ_i, π_i, n_i, h son los numero de electrones sigma, pi, pares de electrones de valencia y número de átomos de hidrogeno respectivamente.

Estos descriptores suelen correlacionar bien con la distribución de cargas en moléculas, y por lo tanto también con los momentos dipolares de hidrocarburos aromáticos e insaturados.[56-59]

3.5.4 Cuenta de trayectos Moleculares

Descriptores moleculares obtenidos contando caminos y trayectos moleculares de ida y de ida y vuelta de diferentes longitudes. Ejemplos de estos son: cuenta de trayecto molecular de orden 1 hasta orden 10.

3.5.5 Indicies de conectividad

Descriptores moleculares calculados a partir del grado del vértice del átomo de un grafo libre de hidrógenos.[60]

3.6 Descriptores 3D

Los descriptores tridimensionales toman en consideración la conformación de la estructura molecular: distancias de enlaces, ángulos de enlaces, ángulos diedros, etc., pudiendo describir entonces las propiedades estereoquímicas de las moléculas. La matriz molecular ($\mathbf{M}$) está formada por las coordenadas cartesianas x, y, z de cada átomo en la geometría molecular optimizada, que se calculan respecto del centro geométrico de la molécula para obtener invariancia en las traslaciones. Por lo general, los descriptores 3D se calculan en base a $\mathbf{M}$ o la matriz distancia geométrica $\mathbf{G}$ (ver sección 2.3).

Para el cálculo de cualquier tipo de descriptor 3D se debe optimizar primero la geometría molecular con algún método apropiado. Habitualmente, esto se realiza con los métodos de Campo de Fuerza de la Mecánica Molecular, MM+, en combinación con métodos derivados de la Mecánica Cuántica, como por ejemplo Hartree-Fock, Métodos

Semiempíricos, u otros. Teniendo en cuenta que, la mejora en la descripción de la estructura trae aparejada un aumento en la demanda computacional y de tiempo de cálculo. (ver sección 2.5.1)

En general, la optimización de geometría se hace con un algoritmo de optimización local, lo cual significa que puede converger a mínimos locales diferentes dependiendo de la geometría inicial. Una aproximación es desarrollar optimizaciones comenzando con un número de conformaciones de partida diferentes y elegir entre ellas la estructura 3D con la menor energía. Se puede recurrir al conocimiento químico para comenzar con conformaciones razonables, que hace que la probabilidad de alcanzar el mínimo global de energía aumente, pero aún así no hay garantía de lograrlo efectivamente, especialmente para moléculas muy grandes. Además se debe tener presente, que no es seguro que la conformación de menor energía sea la más activa en un sistema biológico. Y que los cálculos de las optimizaciones se hacen en vacío pudiendo variar la conformación de mínima energía dependiendo del medio en que se use la sustancia. Esta es la contracara de usar descriptores 3D y que no se presenta en los descriptores tipo 0D, 1D, y 2D. Sin embargo, cómo se verá más adelante, estos problemas son minimizados al usar un pool de descriptores grande; ya que el método de búsqueda no seleccionará aquellos descriptores que no tienen información estructural adecuada.

3.6.1 Perfiles moleculares de Randić

Estos índices se elaboraron especialmente para caracterizar la forma molecular. A pesar de que la forma de una molécula es un concepto importante en Química, el mismo permanece elusivo ya que se lo ha empleado en forma cualitativa a lo largo de los tiempos y sin tratar de definirlo en términos matemáticos precisos. Si bien todos tenemos noción de lo que el término se refiere, su caracterización numérica se vuelve complicada.

Uno de los primeros intentos más simples de describir la forma molecular fue para sistemas similares al benceno, en el que se inscribía la molécula dentro de un rectángulo y se medía sus dimensiones;[61] así se pudo explicar los índices de retención cromatográficos de estos compuestos.

Otras medidas más precisas de la forma se basan en la consideración de la geometría molecular con la matriz distancia geométrica **G**, o en la representación de la molécula con un conjunto de contornos o de superficies de igual densidad electrónica para modelos bidimensionales y tridimensionales, respectivamente.

Los perfiles moleculares de Randić son descriptores moleculares derivados de la distribución de la distancia de la matriz **G**, definidos como el promedio de las sumas por columnas de los elementos elevados a una potencia n, normalizados por el factor n!.[31, 62]

3.6.2 Descriptores RDF

Los descriptores RDF (Funciones de Distribución Radial) se basan en la distribución de las distancias en la geometría molecular. La función de distribución radial de un grupo de átomos pude interpretarse como la probabilidad de encontrar un átomo en un volumen esférico de cierto radio centrado en un punto determinado (en general se centran en diferentes distancias interatómicas de 0.5 A a 15.5 A). Por medio del uso de diferentes esquemas de ponderación atómicos, los RDF pueden ajustarse para obtener las contribuciones atómicas más importantes al descriptor. Estos descriptores proveen de información valiosa acerca de longitudes de enlace, distancias interatómicas, tipos de anillos, información sobre sistemas planos y no-planos, tipos de átomos presentes, y más.[63]

3.6.3 Descriptores WHIM

Las variables WHIM, Invariantes Holísticos Moleculares Pesados, se basan en índices estadísticos calculados durante la proyección de los átomos a lo largo de los ejes principales, con el propósito de capturar información molecular 3-D relevante tal como tamaño molecular, forma, simetría, y distribución atómica con respecto a la referencia invariante. Esto hace que WHIM resulten invariantes a la traslación y a la rotación, o sea, no alteran su valor si la molécula se traslada o rota. Existen descriptores WHIM direccionales y globales, que se diferencian dependiendo si la información estructural se expresa a lo largo de los ejes o si se hace para toda la molécula. La proyección atómica

se logra aplicando el método de Análisis de Componentes Principales a las coordenadas cartesianas centradas, usando la matriz de covarianza pesada. Sus elementos son:

$$s_{jk} = \frac{\sum_{i=1}^{A} w_i.(q_{ij} - \bar{q}_j).(q_{ik} - \bar{q}_k)}{\sum_{i=1}^{A} w_i} \qquad (3.6.1)$$

donde s_{jk} es la covarianza pesada entre las coordenadas atómicas j y k, A es el número de átomos, w_i es el peso del átomo i, q_{ij} y q_{ik} representan las coordenadas del átomo i (j, $k = x, y, z$) y $\bar{q}$ es el correspondiente valor promedio.

La covarianza pesada se pude obtener por diferentes esquemas de pesaje para los átomos: el caso no pesado ($w_i = 1$); masa atómica, volumen de Van der Waals, electronegatividad de Sanderson, Polarizabilidad e índices de estado electro topológicos.[64, 65]

3.6.4 Descriptores 3D MoRSE

Los descriptores 3D MoRSE, Representación Molecular 3D de la Estructura basada en Difracción Electrónica, proveen información 3D de la estructura tridimensional de la molécula usando una transformada derivada de una ecuación usada en estudios de difracción electrónica. Varias propiedades atómicas pueden tenerse en cuenta dada la alta flexibilidad de esta representación de la molécula. La forma simplificada de esta transformada es:

$$I(s) = \sum_{i=2}^{A} \sum_{j=1}^{i-1} w_i.w_j \frac{\text{sen}(s.r_{ij})}{s.r_{ij}} \qquad (3.6.2)$$

$$0 \le s \le 31.0 \text{Å}^{-1}$$

en la cual $I(s)$ es la intensidad electrónica dispersada en una distancia reciproca s, A es el numero de átomos, w es la propiedad atómica elegida como factor de peso (ver sección 3.6.3) la cual permite reflejar la distribución tridimensional de distintas propiedades atómicas en la molécula y r_{ij} es la distancia interatómica entre los átomo i y j. Los valores de s son considerados solo en posiciones discretas dentro de un cierto rango, normalmente son elegidos 32 valores equidistantes entre 0 y 31 Å^{-1}, dependiendo de este rango es la resolución del código que representa la estructura 3D.[66, 67] Los

descriptores 3-D Morse han demostrado describir correctamente la ramificación molecular.

3.6.5 Descriptores GETAWAY

Los índices GETAWAY, Ensamblado de Pesos de Átomos, Geometría y Topología, han sido diseñados con el propósito principal de tratar de conectar la estructura tridimensional de la molécula provista por la matriz de influencia y la información química usando diferentes esquemas de pesos atómicos [63]. La Matriz influencia **H** se define como:

$$\mathbf{H} = \mathbf{M}.(\mathbf{M}^{T}.\mathbf{M}).\mathbf{M}^{T} \qquad (3.6.3)$$

donde **M** es la matriz molecular. La matriz resultante de a x a es invariante respecto a la rotación de las coordenadas moleculares. Los elementos diagonales h_{ij} de la matriz **H** llamados apalancamientos, representan la "influencia" de cada átomo en determinar la forma entera de la molécula. Los átomos que están en la parte externa de la molécula tienen mayores h_{ij} respecto de los más cercanos al centro molecular. Además, la magnitud del máximo h_{ij} es dependiente del tamaño y forma de la molécula. Moléculas que tienen simetría esférica tienen valores bajos de h_{ij}, mientras que aquellas que son lineales poseen valores más altos. En moléculas que poseen formas similares, el apalancamiento máximo decrece a medida que aumenta su tamaño (número de átomos). Cada elemento fuera de la diagonal h_{ij} representa el grado de accesibilidad del átomo j a interacciones con otro i. Valores negativos de h_{ij} indicarían que los dos átomos ocupan regiones moleculares opuestas respecto del centro, por lo que la accesibilidad mutua será baja. Cabe mencionar que átomos equivalentes tendrán iguales apalancamientos.

3.6.6 Descriptores Geométricos

Diferentes tipos de descriptores que se calculan a partir de la geometría molecular: longitudes de enlace (determinadas por los tipos de átomos y la multiplicidad), ángulos de enlace, ángulos diedros, etc. Ejemplos de descriptores geométricos son los índices gravitacionales,[68] Índice de Wiener tridimensional,[69] la suma de distancias geométricas entre pares de átomos, etc.

3.6.6.1 Descriptores Aromáticos

Una clase importante de descriptores geométricos son los Aromáticos. A pesar de que al término aromaticidad se lo emplea de forma universal, no es una cantidad medible de forma directa. Sin embargo, se acepta caracterizar este fenómeno a partir de varios modelos. El índice HOMA (modelo de oscilador armónico de la aromaticidad) representa la tendencia de la longitud enlace a estar comprendida entre la longitud de un enlace simple y uno doble.[70]

3.6.7 Descriptores de carga

Los descriptores de carga describen la distribución de cargas en la molécula, y como tales solo son confiables cuando la molécula ha sido optimizada con algún método mecano cuántico adecuado que permita determinar las cargas atómicas.[71] Ejemplos de estos son la suma de las cargas atómicas, carga atómica positiva máxima, carga atómica negativa máxima, carga positiva total, carga cuadrática total, energías HOMO y LUMO, etc.

Las energías HOMO-LUMO, no son entregadas por el software de cálculo de descriptores usado y pueden resultar de gran utilidad incluirlas en los estudios QSPR-QSAR. Sin embargo estas pueden ser extraídas del archivo de salida del Hyperchem[49] (ver sección 2.5.1), debido a que esto puede ser una tarea muy laboriosa cuando el número de moléculas es alto, lo que es usual en los estudios QSAR, y que puede acarrear muchos errores humanos, se realizó un programa de automatización programado en Excel,[72] el cual se denomina "Logs.xls" y queda disponible en las computadoras del grupo. Cabe mencionar que el código de Excel no es en forma de líneas de texto por lo que no es posible agregarlo en el Apéndice.

El descriptor HOMO corresponde al orbital molecular más alto ocupado con electrones, mientras que LUMO es la energía del orbital molecular más bajo desocupado. Ambos tipos de energía caracterizan la reactividad de un compuesto. La primera es indicadora de la susceptibilidad de una molécula hacia el ataque por electrófilos mientras que la segunda expresa la tendencia de ser atacada por nucleófilos. Por ejemplo, cuando una molécula actúa como base de Lewis en la formación de un enlace los electrones puestos en juego corresponden al orbital HOMO, y la facilidad

para que ello ocurra dependerá del valor de energía; si es un ácido de Lewis, entonces la molécula recibirá los electrones en el orbital LUMO. La diferencia de energías HOMO-LUMO puede relacionarse con la estabilidad química de compuestos. Una alta diferencia energética es indicador de una mayor estabilidad de un determinado compuesto hacia una reacción química, especialmente en el caso de reacciones radicalarias.

3.7 Consideraciones respecto a descriptores noveles

Se ha mencionado en la literatura que se está buscando un descriptor novel, es decir aquel que englobe toda la información estructural de la molécula en una única variable y que a su vez sea aplicable a cualquier propiedad-actividad bajo estudio[37], y que debido a la complejidad de la estructura de una molécula cualquiera hasta el presente no se ha logrado encontrarlo. La opinión del autor es que esto no sería posible ni necesario. Un descriptor, que finalmente es un número, no puede ser correlacionado con diferentes propiedades ya que para que esto sea posible las propiedades deberían tener los mismos valores. Adicionalmente, tal como se verá más adelante en la aplicación de las teorías QSAR/QSPR a distintos problemas (ver sección 6), las correlaciones obtenidas usando varios descriptores (y no un único descriptor novel) es óptima y se pude llevar a cabo sin ningún inconveniente. Estos grupos de descriptores representan distintos aspectos estructurales que varían para las distintas propiedades permitiendo además interpretar cuales son los factores estructurales que afectan la propiedad en estudio. Esto no quiere decir que sea inútil continuar investigando distintas alternativas de descriptores, ya que es trascendental obtener nuevos descriptores que optimicen la codificación de algún aspecto particular de la molécula como; por ejemplo: la polarizabilidad, el grado de nucleofilicidad o electofilicidad.

4 Diseño de Modelos

"La suerte favorece a la mente preparada" (Louis Pasteur)

4.1 Introducción

El diseño de un modelo involucra dos etapas fundamentales; la selección de los descriptores que formarán parte del modelo y la búsqueda de la función matemática que se usará en el mismo, la cual puede ser lineal o no-lineal.

Por lo general, suelen hablarse de tres tipos de diseños: modelos tipo I, II y III. En los modelos tipo I, la selección de los descriptores y función se hace con regresiones lineales múltiples (MLR), es decir, son completamente lineales. Los modelos tipo II seleccionan los descriptores por medio de MLR y usan una función no-lineal a través de Redes Neuronales Artificiales (ANN, del inglés Artificial Neural Networks). Por último los modelos tipo III, seleccionan los descriptores y la función de forma no lineal (ANN). Estos tres casos se pueden resumir en la Tabla 4.1.1.

Tabla 4.1.1 Clasificación de modelos

Modelo tipo	Selección de descriptores	Función matemática
I	lineal	lineal
II	lineal	no-lineal
III	no-lineal	no-lineal

4.2 Métodos de búsqueda

Los métodos de búsqueda o selección de descriptores son usados para encontrar un subconjunto óptimo dentro del conjunto completo de descriptores, este conjunto optimo estará conformado por los descriptores que formarán el modelo. Como se mencionó en la sección 1.4 en la actualidad existen miles de descriptores disponibles por lo que una búsqueda completa sería impracticable. Como veremos más adelante existen diversos métodos aproximados de búsqueda que optimizan la selección del subconjunto.

Asimismo tal como se mencionara en la sección 1.4, es una regla aceptada en la práctica que al establecer un modelo matemático debe comenzarse desde lo más simple, y sólo si es necesario, proceder a un nivel más complejo en su formulación matemática. Teniendo esto en cuenta en este trabajo de tesis se emplearon solo métodos lineales tipo I, ya que optimizando los mismos se logran excelentes resultados, que han mostrado ser comparables con los resultados de métodos no lineales siendo estos mucho más sofisticados como ANN[73]. Cabe mencionar que una importante desventaja de los métodos no lineales, es su carácter de "caja negra" donde solo se ingresan variables y se obtienen resultados de salida[27], haciéndose imposible interpretar la relación entre la estructura y la actividad; como contracara los métodos que se basan en regresiones lineales permiten analizar e interpretar los modelos obtenidos sin inconvenientes, siendo esta otra razón por la cual se eligieron estos últimos para ser empleados en este trabajo de tesis.

4.2.1 Problema a resolver

El problema a ser resuelto consiste en encontrar d variables (descriptores) óptimas $\mathbf{d}=\{X_1,X_2,X_3,...,X_d\}$ que dan origen al modelo a partir de un conjunto de descriptores $D>>d$. Por conjunto óptimo se entiende como aquel que produce los mejores parámetros estadísticos, es decir desviación estándar (S) y coeficiente de correlación (R).

Si se elige el mínimo valor de S como criterio de búsqueda, entonces aquel conjunto de d variables con menor S será la mejor solución posible dentro del conjunto inicial analizado. En términos matemáticos, debemos encontrar el mínimo global de $S(\mathbf{d})$ donde $\mathbf{d}$ es un punto en un espacio de $D!/[d!(D-d)!]$ puntos.

Cabe mencionar que este problema no puede tratarse como una minimización habitual donde en general lo que se busca es el mínimo de una función en una hipersuperficie que presenta puntos ordenados respecto a un origen de coordenadas. En el caso del conjunto de descriptores D no existe un orden y por lo tanto no se puede establecer una distancia entre dos puntos cualesquiera, esto hace más complicado cualquier análisis.

4.2.2 Método de búsqueda exhaustiva (FS)

El método de búsqueda exhaustiva o en inglés Full Search (FS) es aquel donde se prueban todos los casos posibles para encontrar el mejor modelo de d descriptores a partir de un número mucho mayor D, para esto es necesario explorar todo el espacio y por consiguiente realizar $D!/[d!(D-d)!]$ regresiones lineales.

Para tener noción de lo que significa llevar a cabo una búsqueda exhaustiva en términos computacionales se estimará el tiempo necesario usando como ejemplo la aplicación con menor cantidad de descriptores mostrada en la sección 6.4. Para este caso D=1057 y el modelo consta de cuatro descriptores (d=4) lo que da un valor de $D!/[d!(D-d)!]=5.2\text{x}10^{10}$. Para analizar este número de casos con un procesador Athlon 64 2800+ usando Matlab tardaría aproximadamente 900000 años. (Para más de talles de cómo se pude hacer esta estimación por favor remitirse a la sección 7.3). Y de usarse d=7, siendo este un número razonable en aplicaciones QSAR, se necesitarían 2.8 $\text{x}10^{19}$ años. Estos números exuberantes indican que aún con una computadora varias veces más potente no se llegarán a realizar los cálculos en un tiempo razonable.

Por lo tanto es necesario emplear métodos aproximados; en la literatura especializada existe una enorme cantidad de este tipo de algoritmos y en este trabajo solo se mencionaran aquellos que consideramos relevantes.

4.2.3 Método de Regresión "paso a paso" (FSR)

El método de regresión paso a paso o en ingles Forward Stepwise Regresion (FSR)[74] se usa desde hace mucho tiempo y es clásico en los trabajos QSPR-QSAR. Su popularidad se debe a tratarse de un procedimiento rápido, sencillo, con un costo computacional mínimo, y que se halla disponible en cualquier paquete computacional comercial.

La inclusión paso a paso consiste en calcular en una primer etapa el mejor modelo de una variable, luego en cada etapa subsiguiente adicionar una nueva variable que mejore la calidad del modelo, y el modelo óptimo se encuentra cuando no es posible mejorar más la relación.

El método de regresión paso a paso no garantiza que la solución óptima coincida con la solución exacta, simplemente porque cada vez que se introduce/remueve una

variable en el modelo, ya hay otras presentes en el mismo que restringen y determinan la calidad de la ecuación.

Aquí aparece lo que se conoce con el nombre de "efecto de mezcla de variables", resultante de que las variables se combinan entre sí para producir un determinado efecto final sobre los parámetros estadísticos. El uso de una regresión de a pasos no tiene en cuenta dicho efecto, y esto explica que el método no alcance resultados suficientemente buenos. Una ventaja de este método es que no necesita un grupo de descriptores como punto de partida.

4.2.4 Algoritmos Genéticos (AG)

Un Algoritmo Genético (AG) [75] es una técnica de búsqueda basada en la evolución natural, donde las variables tienen el rol de genes (para el caso de QSAR/QSPR esto serán los descriptores) en un individuo (en QSAR/QSPR esto será un modelo). Un conjunto inicial de individuos (población) evoluciona de acuerdo a una función de robustez (desviación estándar en QSAR/QSPR) que determina la supervivencia de los individuos.

El algoritmo busca aquellos individuos que dan mejores valores de la función de robustez mediante los operadores genéticos de selección, mutación y entrecruzamiento. El operador de selección garantiza la propagación de los individuos con mayor resistencia hacia poblaciones futuras. Los AG exploran el espacio de soluciones combinando genes de dos individuos (padres) usando el operador de entrecruzamiento para formar dos nuevos individuos (hijos) y además mutando aleatoriamente individuos usando el operador de mutación. Estos algoritmos ofrecen una combinación de la habilidad de ascenso de lomas (selección natural) y la de un método estocástico (entrecruzamiento y mutación) explorando muchas soluciones en paralelo procesando la información de una manera muy eficiente.

La aplicación de los AG en la práctica requiere el ajuste de algunos parámetros como tamaño de la población, brecha generacional, probabilidad de entrecruzamiento y probabilidad de mutación. Estos parámetros típicamente interactúan de manera no lineal y no pueden ser optimizados uno a la vez. Esto hace que se plantee un problema adicional difícil de resolver. Hay mucho material respecto a el ajuste de parámetros, donde se presentan distintas formas de encarar el problema y discusiones al respecto; sin embargo no hay resultados concluyentes acerca de la mejor forma de ajustarlos. [76]

4.2.5 Método de Reemplazo (RM)

El Método de Reemplazo (RM) [77-79] surgió hace un tiempo debido a la carencia de un programa computacional comercial que permita analizar cientos de descriptores moleculares a fin de encontrar los mejores subconjuntos de variables. El método provee una herramienta aplicable a cualquier estudio QSPR-QSAR.

La principal ventaja de RM es que requiere un número de regresiones lineales mucho menor al de una búsqueda exhaustiva y produce resultados finales cercanos a los exactos. Se trata de una aproximación que tiene en cuenta el efecto de mezcla de variables comentado en la sección 4.2.3.

La idea principal de RM es que se puede lograr un valor pequeño de S tomando en cuenta la desviación estandard relativa (*der*) de los coeficientes del ajuste lineal con d descriptor es $\mathbf{d_m}=\{X_{m1},X_{m2},X_{m3},...,X_{md}\}$. El fundamento de emplear este camino para efectuar la búsqueda de variables radica en la simple observación de que los modelos obtenidos haciendo una busqueda exhaustiva producen *der* bajas en los coeficientes. En la actualidad no sabemos de otro método reportado en la literatura que también este basado en *der*. La razón tal vez sea que experimentos numéricos conducidos tiempo atrás por otros investigadores sugirieron que los coeficientes de regresión y sus errores asociados exhibían un comportamiento aleatorio, determinado tal vez por la inestabilidad de las ecuaciones de regresión.

La esencia del procedimiento RM es la siguiente: primero se elige un conjunto $\mathbf{d}_k$ de manera aleatoria y se hace una regresión lineal. Luego se escoge uno de los descriptores del conjunto, digamos X_{ki}, se lo reemplaza por cada uno de los $D-d$ descriptores restantes del conjunto total , y se conserva el conjunto resultante con menor S ; a esto lo llamaremos un paso.

Debido a que en este primer paso se puede comenzar con cualquiera de los d descriptores en el modelo inicial, se tendrán d caminos distintos que conducen a soluciones finales.

Luego se elige la variable en el modelo resultante que posea la mayor *der* en su coeficiente (omitiendo la que fue reemplazada en el paso previo) y se reemplaza por los $D-d$ descriptores restantes, reteniendo nuevamente el mejor conjunto resultante. De ese modo se reemplazan las variables remanentes omitiendo las reemplazadas en pasos previos. Una vez finalizado este ciclo de d pasos se comienza nuevamente con la variable

que tiene la mayor *der* en su coeficiente y se repite el ciclo completo. Este proceso se repite tantas veces como sea necesario hasta que el conjunto de descriptores resulte invariante. Al final, tendremos el mejor modelo para el camino *i*. Se procede exactamente de la misma manera para todos los caminos posibles $i = 1,2,\ldots,d$, y luego se elige el conjunto de descriptores con menor *S*.

El Método de Reemplazo es un algoritmo iterativo de muy rápida convergencia que produce modelos con *S* pequeña en un tiempo de cálculo notablemente reducido. [79-81] Sin embargo, en algunos casos, el RM puede quedar atrapado en un mínimo local de *S* del que no pude escapar. A pesar de que estos mínimos locales proveen modelos aceptables, como lo muestra las aplicaciones de RM [79-81], hay todavía espacio para mejoras.

4.2.6 Método de Reemplazo Modificado (MRM)

El Método de Reemplazo Modificado (MRM)[82] sigue la misma filosofía que el RM aunque muestra menor predisposición a permanecer en un mínimo local y a la vez es menos dependiente de la solución inicial.

Este nuevo algoritmo surgió como consecuencia del presente trabajo de tesis y tiene la misma forma que el algoritmo RM excepto que en cada paso se sustituye al descriptor con el error más alto aún si la sustitución no produce un valor menor de *S* (se elige el próximo menor valor de *S*). El algoritmo converge a diferentes soluciones y comúnmente rebota de un punto a otro, repitiendo ocasionalmente algunos de ellos; en esos casos se encontró que una solución aceptable es la que primero se repite cuatro veces. Si la convergencia no se logra en un tiempo aceptable se detiene el proceso luego de 350 pasos, esto no conlleva una pérdida importante ya que los *S* de los modelos resultantes siempre son lo suficientemente pequeños.

4.2.7 Método de Reemplazo Ampliado (ERM)

Al comparar MRM con RM se observó que el primero pareciera tener una agitación térmica mayor que el segundo. Esto llevó a intentar integrar los algoritmos en distintas combinaciones alternando RM y MRM. La combinación óptima encontrada fue: RM-MRM-RM a la que se denominó ERM (Enhanced Replacement Method)[82]; esto está en

línea con el hecho de que tal combinación se podría asemejar a un algoritmo de Simulación de Recocido o en inglés Simulated Annealing (SA), el cual es una adaptación del algoritmo de Metropolis-Hastings, un método de Monte Carlo [83] para generar estados de muestra de un sistema termodinámico. El nombre e inspiración vienen del recocido en metalúrgica, una técnica que involucra el calentamiento y el enfriamiento controlado para incrementar el tamaño de los cristales y reducir sus defectos. El calentamiento hace que los átomos se "destraben" de su posición inicial (un mínimo local de energía interna) y se muevan aleatoriamente por niveles de mayor energía; luego el enfriamiento lento les da más chances de encontrar configuraciones con menor energía interna que la inicial. [84].

Los resultados obtenidos por este nuevo algoritmo (ERM), el cual también surgió como resultado de este trabajo de tesis, son superiores a los de MRM y RM por separado.

Adicionalmente más adelante se verán variaciones de ERM para mejorar el mismo.

4.3 Comparación numérica entre RM, MRM y ERM

4.3.1 Introducción y desarrollo

A continuación se mostraran los ensayos numéricos que se llevaron a cabo de forma de probar y comparar el desempeño de RM y MRM. El análisis de los datos luego derivó en la obtención de ERM. Para llevar a cabo tales ensayos se usaron cuatro conjuntos de datos experimentales distintos que ya habían sido analizados previamente: un conjunto de datos de fluorofilicidad (FLUOR), el cual contiene 116 compuestos orgánicos caracterizados por 1268 descriptores teóricos [80]; un conjunto de datos de Inhibición del Crecimiento (GI) de *Tetrahymena pyriformis* por 200 fenoles con 1338 descriptores,[81], un conjunto de datos de receptores GABA (GABA), que contiene 78 datos de inhibición para derivados de flavona con 1187 descriptores[85] y un conjunto de 100 datos de ED_{50} (MES) con 1306 descriptores[86-88]. Adicionalmente un conjunto de datos de 209 Bifenoles policlorados (PCB) con medidas del factor de respuesta relativa con 63912 descriptores moleculares [89] se usó para determinar si era posible la aplicación del nuevo algoritmo a bases de datos extremadamente grandes.

Con el propósito de poder visualizar el comportamiento del nuevo algoritmo MRM y compararlo con RM, las Figuras 4.3.1 y 4.3.2 muestran el descenso de S en función del

número de pasos para una optimización de siete parámetros para la base de datos de FLUOR [80], usando RM y MRM respectivamente. De los gráficos se desprende que MRM pude asemejarse a un proceso con mayor agitación térmica o ruido que RM, tal como se mencionó anteriormente. A pesar de su mayor temperatura aparente, la tendencia de minimizar S se mantiene, haciendo que MRM sea menos propenso a caer en un mínimo local a expensas de una convergencia más lenta y un costo computacional mayor.

El análisis de estos gráficos llevó a pensar en distintas alternativas que integren los dos algoritmos: se probaron las siguientes combinaciones: MRM-RM, RM-MRM y RM-MRM-RM. Cabe mencionar que en el caso de MRM-RM, la solución de partida de RM es la mejor solución encontrada al aplicar MRM. La combinación MRM-RM-MRM fue descartada por incrementar significativamente el tiempo de cálculo sin mostrar resultados que lo justifiquen. En las Figuras 4.3.3, 4.3.4 y 4.3.5 se puede observar la variación de S en función del número de pasos para las combinaciones mencionadas, usando nuevamente el conjunto de datos de FLUOR.

De forma tal de poder comparar los algoritmos antes mencionados con una búsqueda exhaustiva, se seleccionaron descriptores moleculares de los cuatro conjuntos totales reduciendo el conjunto **D** en cada caso a D=75 variables para de esta forma poder llevar a cabo una FS en tiempos razonables. Luego se aplicaron los algoritmos y se obtuvieron modelos óptimos de d= 1,2,...,7 descriptores usando la misma solución inicial aleatoria en cada caso. Todos los modelos incluyen el término constante. Los resultados se resumen en la Tabla 4.3.1 que compara los valores de S obtenidos por todos los algoritmos y para mayor claridad los valores mínimos exactos se muestran en negrita.

De la Tabla 4.3.1 se puede observar que S_{RM} es igual o similar a S_{FS}. Además se puede ver que MRM entrega mejores resultados que RM y que estos pueden ser aún mejores usando las diferentes alternaciones; de hecho la combinación RM-MRM-RM parece dar los mejores resultados. El número de regresiones lineales (mostrado en paréntesis al final de la Tabla 4.3.1) requeridas para los algoritmos alternativos es mayor a RM pero permanecen menores que FS para valores de d menor a 2.

Luego se llevaron a cabo pruebas de los nuevos algoritmos en problemas reales usando las cuatro bases de datos completas. En este caso no es posible llevar a cabo una FS en un tiempo razonable, como ya se mostró en la sección 4.2.2. Un colega nos propuso una alternativa para comparar los distintos algoritmos frente a la solución exacta sin tener que llevar a cabo una FS, modificando la base de datos original con el fin de agregar una solución exacta conocida de antemano. Sin embargo al realizar diversas pruebas los

resultados mostraron que al forzar esta solución el problema era alterado notablemente simplificándose demasiado lo que impidió poder comparar los distintos algoritmos, los detalles de estas pruebas se pueden ver en la sección 8.2 del Apéndice.

Los ensayos numéricos se hicieron usando d=7 para tener un problema con alto nivel de exigencia computacional, siendo además un número razonable de descriptores en estudios QSAR/QSPR. Como normalmente los modelos óptimos aproximados dependen del grupo inicial de descriptores elegidos, se usaron los mismos tres grupos aleatorios en todos los algoritmos, los resultados se muestran en la Tabla 4.3.2.

La última columna de la Tabla 4.3.2 muestra además los resultados provistos por el método paso a paso (FSR, para detalles referirse a la sección 4.2.3) como punto de comparación. La Tabla 4.3.2 muestra además el promedio del porcentaje de mejoras respecto a RM para facilitar la visualización del desempeño de los algoritmos. Al final de la tabla se ve el cociente del número de regresiones lineales para los nuevos algoritmos respecto a RM.

De la Tabla 4.3.2 se pude ver que RM presenta mejores resultados que FSR, confirmando estudios comparativos previos [80]. Además se observa que MRM tiene mejor o igual desempeño que RM para todos los casos excepto para una solución inicial del conjunto de datos de FLUOR. Este caso particular aparenta ser fortuito, ya que MRM es claramente superior a RM para las dos soluciones de partida restantes de la misma base de datos. Debe tenerse en mente que los resultados de los métodos aproximados dependen de la solución de partida usada y por lo tanto siempre es posible que un método de valores más pequeños de S que otro algoritmo mejor. Inclusive hay una baja probabilidad pero no igual a cero de que un algoritmo más pobre pueda acertar al mínimo de S global.

El porcentaje de mejora de la Tabla 4.3.2 sugiere que las combinaciones de algoritmos propuestas dan mejores resultados que MRM. En particular la secuencia RM-MRM-RM (ERM) emerge como el mejor algoritmo; esto está en línea con la idea de que es el único algoritmo que fue sometido a un ciclo completo de simulación de recocido[84], cómo se aprecia en la Figura 4.3.5. La demanda computacional de ERM es comparable a MRM y es casi siete veces mayor a la de RM. Este es el precio a pagar para obtener modelos QSAR/QSPR con parámetros estadísticos mejores a los obtenidos en el pasado [80].

La Tabla 4.3.2 sugiere que los resultados de ERM son menos sensibles a la solución inicial en comparación con RM. Sin embargo todavía existe una dependencia entre los

resultados obtenidos y la solución inicial usada para ERM y este es un punto que será tratado más adelante.

Para probar el ERM en un problema mucho más demandante, se hicieron pruebas en el conjunto de datos de PCB que contiene 63912 descriptores. ERM convergió en un tiempo razonable, los resultados se muestran en la Tabla 4.3.3. Como era de esperar ERM dio valores de S más bajos que RM para los mismos puntos aleatorios de partida.

4.3.2 Conclusiones

Se encontró un algoritmo que mejora los resultados de RM el cual fue llamado MRM, así mismo se obtuvieron algoritmos compuestos que se asemejan a un simulado de recocido. El algoritmo más eficiente resultó ser el ERM=RM-MRM-RM el cual entrega modelos con mejores parámetros estadísticos y es menos sensible al punto de partida del procedimiento. Si bien este algoritmo presenta mayor demanda computacional esto no parece contrarrestar sus ventajas como se comprobó en la aplicación del mismo en problemas reales y en la prueba con un requerimiento computacional extremo realizando la búsqueda entre 63912 descriptores.

Tabla 4.3.1 Desviación estándar (S) y número de regresiones lineales para: FS, RM, MRM, RM-MRM, MRM-RM y ERM (RM-MRM-RM), para cuatro subconjuntos de datos de D=75 descriptores. La barra " / " separa algoritmos que dan resultados idénticos. Los resultados FS se muestran en negrita.

Algoritmo	S						
d	1	2	3	4	5	6	7
MES							
FS	**0.3991**	**0.3666**	**0.3536**	**0.3443**	**0.3361**	**0.3254**	**0.3169**
RM	**0.3991**	**0.3666**	**0.3536**	0.3480	**0.3361**	0.3327	0.3268
MRM / MRM-RM	**0.3991**	**0.3666**	**0.3536**	**0.3443**	**0.3361**	0.3290	**0.3169**
RM-MRM / ERM	**0.3991**	**0.3666**	**0.3536**	**0.3443**	**0.3361**	**0.3254**	**0.3169**
GI							
FS	**0.6494**	**0.6000**	**0.5693**	**0.5605**	**0.5487**	**0.5324**	**0.5214**
RM	**0.6494**	**0.6000**	0.5875	**0.5605**	0.5512	0.5415	0.5350
MRM	**0.6494**	**0.6000**	**0.5693**	**0.5605**	0.5492	**0.5324**	**0.5214**
RM-MRM	**0.6494**	**0.6000**	0.5875	**0.5605**	0.5492	0.5350	**0.5214**
MRM-RM	**0.6494**	**0.6000**	**0.5693**	**0.5605**	0.5492	**0.5324**	**0.5214**
ERM	**0.6494**	**0.6000**	0.5875	**0.5605**	0.5492	**0.5324**	**0.5214**
FLUOR							
FS	**1.1192**	**0.7587**	**0.7294**	**0.6901**	**0.6440**	**0.6200**	**0.5971**
RM	**1.1192**	0.7891	0.7329	**0.6901**	0.6549	0.6451	0.6253
MRM/Resto	**1.1192**	**0.7587**	**0.7329**	**0.6901**	**0.6440**	**0.6200**	**0.5971**
GABA							
FS	**0.8289**	**0.7335**	**0.6421**	**0.5918**	**0.5719**	**0.5383**	**0.5083**
RM	**0.8289**	**0.7335**	**0.6421**	**0.5918**	**0.5719**	0.5398	0.5120
MRM	**0.8289**	**0.7335**	**0.6421**	**0.5918**	**0.5719**	**0.5383**	0.5088
RM-MRM	**0.8289**	**0.7335**	**0.6421**	**0.5918**	**0.5719**	0.5398	0.5120
MRM-RM / ERM	**0.8289**	**0.7335**	**0.6421**	**0.5918**	**0.5719**	**0.5383**	**0.5083**
Número de regresions lineales promedio							
FS	75	2,775	67,525	1.22E+06	1.73E+07	2.01E+08	1.98E+09
RM	75	1,260	2,850	4,756	7,638	10,086	14,739
MRM	75	4,674	16,095	89,464	81,380	110,958	240,149
RM-MRM / MRM-RM	75	5,934	18,945	94,220	89,018	121,044	254,888
ERM	75	7,194	21,795	98,976	96,655	131,130	269,627
Tiempo de cálculo promedio en minutos*							
FS	2.07E-04	7.67E-03	1.87E-01	3.36E+00	4.77E+01	5.57E+02	5.49E+03
RM	2.07E-04	3.48E-03	7.88E-03	1.31E-02	2.11E-02	2.79E-02	4.07E-02
MRM	2.07E-04	1.29E-02	4.45E-02	2.47E-01	2.25E-01	3.07E-01	6.64E-01
RM-MRM / MRM-RM	2.07E-04	1.64E-02	5.24E-02	2.60E-01	2.46E-01	3.35E-01	7.05E-01
ERM	2.07E-04	1.99E-02	6.02E-02	2.74E-01	2.67E-01	3.62E-01	7.45E-01

*Usando un procesador AMD Athlon 64 2800+

Tabla 4.3.2 Desviación estándar (S), R de la validación *Leave-One-Out* (entre paréntesis), número de regresiones lineales y tiempo de cálculo (entre paréntesis) para RM, MRM, RM-MRM, MRM-RM, ERM (RM-MRM-RM), y FSR, para los cuatro conjuntos de datos completos. Se usaron tres soluciones de partida diferentes de siete descriptores. Las mejores soluciones aparecen en negrita.

Algoritmo	RM	MRM	RM-MRM	MRM-RM	ERM	FSR
			$S\ (R_{loo})$			
MES	0.3089 (0.685)	**0.2896** **(0.726)**	0.2919 (0.722)	**0.2896** **(0.726)**	0.2919 (0.722)	0.3209 (-----)
	0.3077 (0.692)	**0.2973** **(0.722)**	**0.2973** **(0.722)**	**0.2973** **(0.722)**	**0.2973** **(0.722)**	
	0.3008 (0.695)	0.2954 (0.710)	**0.2896** **(0.726)**	0.2951 (0.710)	**0.2896** **(0.726)**	
GI	**0.4421** **(0.835)**	**0.4421** **(0.835)**	**0.4421** **(0.835)**	**0.4421** **(0.835)**	**0.4421** **(0.835)**	0.4937 (0.789)
	0.4648 (0.821)	0.4421 (0.835)	**0.4367** **(0.837)**	0.4421 (0.835)	**0.4367** **(0.837)**	
	0.4445 **(0.835)**	**0.4445** **(0.835)**	**0.4445** **(0.835)**	**0.4445** **(0.835)**	**0.4445** **(0.835)**	
GABA	0.4465 (0.891)	**0.4045** **(0.91)**	0.4269 (0.898)	**0.4045** **(0.91)**	0.4142 (0.903)	0.46797 (0.876)
	0.4683 (0.878)	**0.3961** **(0.912)**	0.4121 (0.903)	**0.3961** **(0.912)**	**0.3961** **(0.912)**	
	0.4160 (0.905)	**0.3961** **(0.912)**	**0.3961** **(0.912)**	**0.3961** **(0.912)**	**0.3961** **(0.912)**	
FLUOR	0.4936 (-----)	0.4572 (0.981)	0.4339 (0.983)	0.4470 (0.982)	**0.4328** **(0.983)**	0.5718 (0.970)
	0.4328 **(0.983)**	0.4647 (0.981)	**0.4328** **(0.983)**	0.4606 (0.981)	**0.4328** **(0.983)**	
	0.4985 (0.976)	0.4426 (0.983)	0.4619 (0.979)	**0.4408** **(0.983)**	0.4470 (0.982)	
Mejora Promedio	**0%** **(0%)**	**4.76%** **(1.8%)**	**4.94%** **(1.8%)**	**5.05%** **(1.9%)**	**5.73%** **(2.0%)**	**-10.31%** **(-2.5%)**
Número de regresiones lineales (Tiempo de cálculo en minutos*)						
Promedio	283878 (0.78)	1629938 (4.51)	1725873 (4.77)	1828165 (5.05)	1926775 (5.33)	8923 (0.02)
Cociente	1	5.74	6.08	6.44	6.79	0.031

*Usando un procesador AMD Athlon 64 2800+

Tabla 4.3.3 Desviación estándar (S), R de la validación *Leave-One-Out* (entre paréntesis), número de regresiones lineales y tiempo de cálculo (entre paréntesis) para RM y ERM para el conjunto de datos PCB con D = 63912. Se usaron tres soluciones de partida diferentes de siete descriptores. Las mejores soluciones aparecen en negrita.

Algoritmo	RM	ERM
S (R_{loo})		
PCB	**0.1616 (0.883)**	**0.1616 (0.883)**
	0.1718 (0.866)	**0.1616 (0.883)**
	0.1616 (0.883)	**0.1610 (0.884)**
Mejora promedio	**0% (0%)**	**2.1% (0.7%)**
Número de regresiones lineales (Tiempo de cálculo en minutos *)		
Promedio	1.42E+07 (39.25)	7.42 E+07 (205.18)
Cociente	1	5.23

*Usando un procesador AMD Athlon 64 2800+

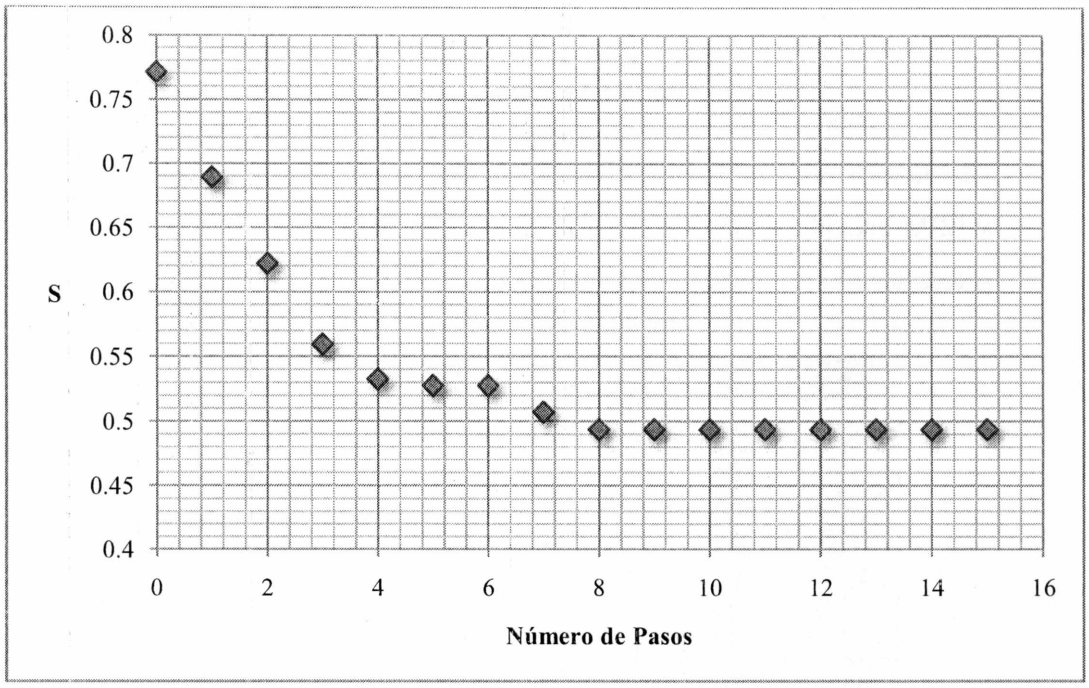

Figura 4.3.1 Desviación Estándar vs. Número de Pasos para RM

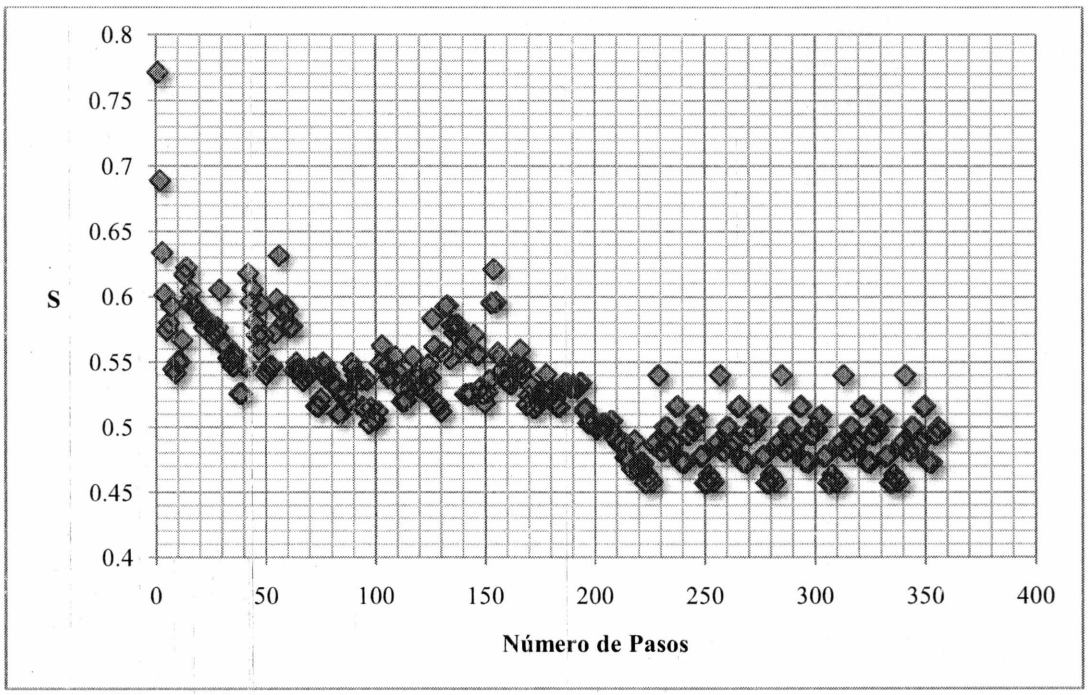

Figura 4.3.2 Desviación Estándar vs. Número de Pasos para MRM

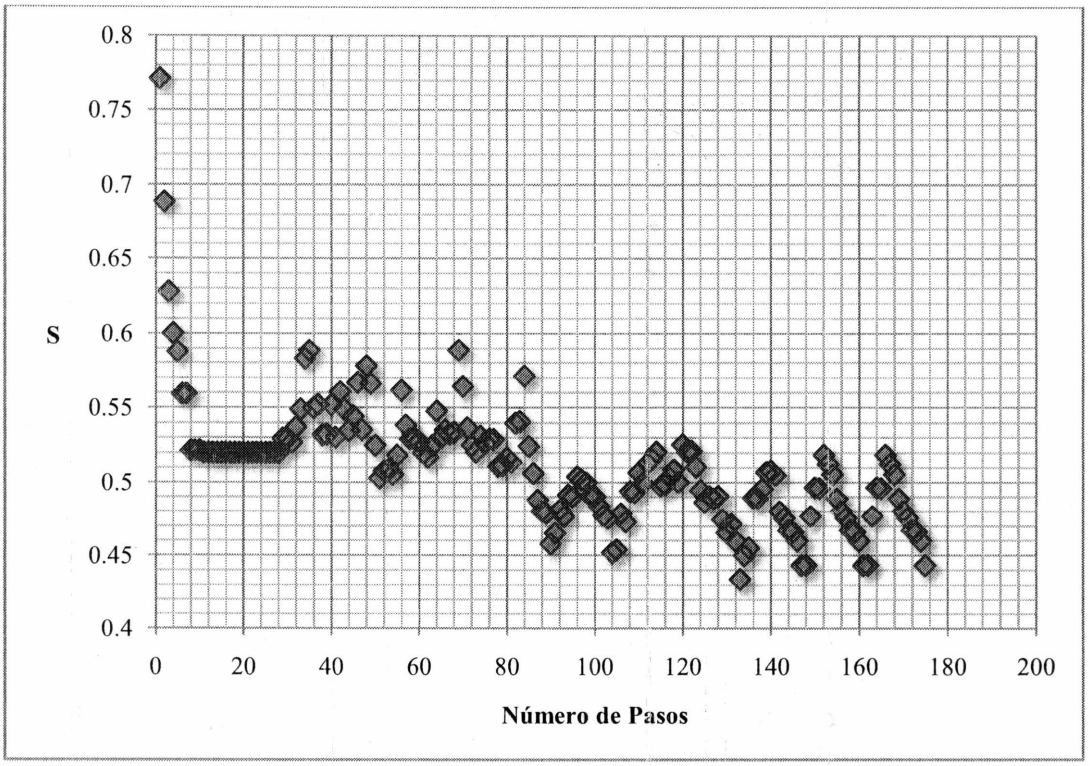

Figura 4.3.3 Desviación Estándar vs. Número de Pasos para RM-MRM

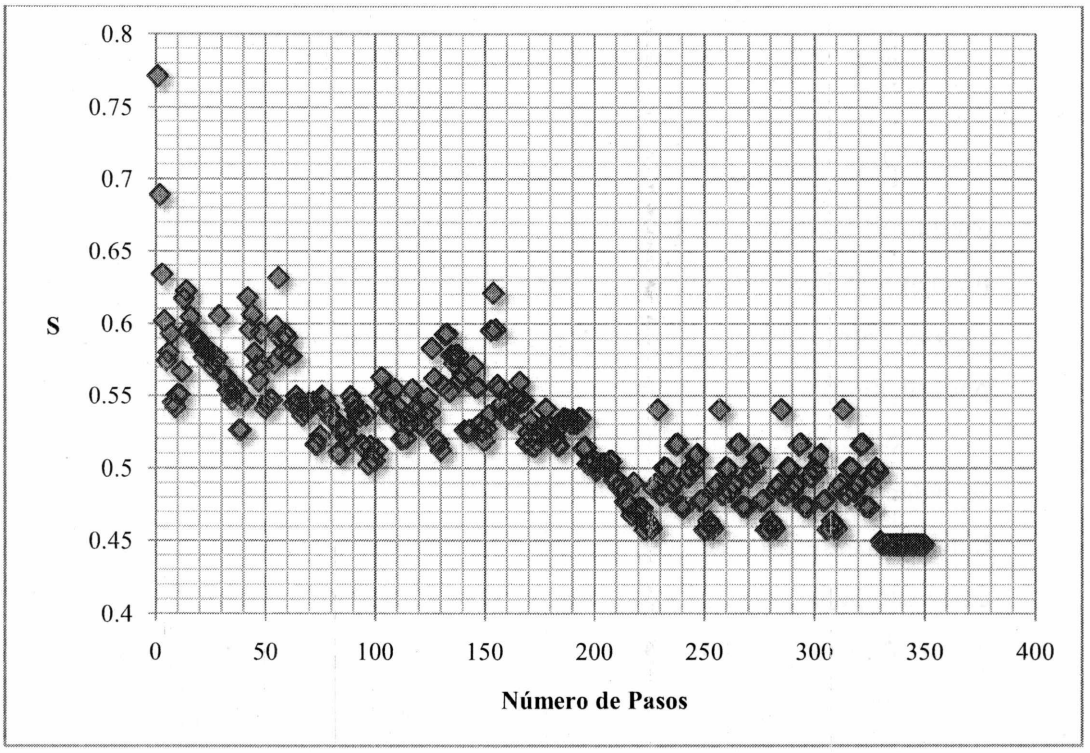

Figura 4.3.4 Desviación Estándar vs. Número de Pasos para MRM-RM

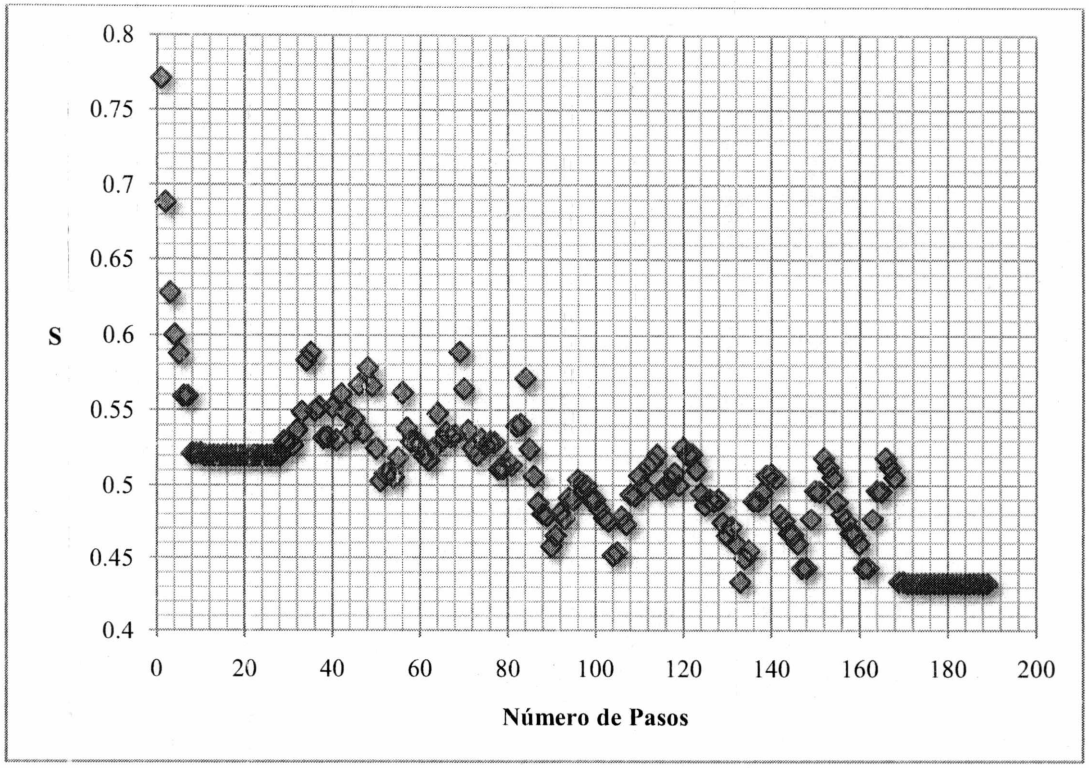

Figura 4.3.5 Desviación Estándar vs. Número de Pasos para ERM (RM-MRM-RM)

4.4 Ensayos en el primer paso de RM

El primer paso de RM no tiene en cuenta la desviación estándar relativa (*der*), en cambio se siguen todos los *d* caminos posibles. Esto es debido a que se notó en el uso práctico del algoritmo que los mejores resultados no siempre dependían del *der* del camino seguido. Sin embargo existían otras alternativas que podrían dar aún mejores resultados; por este motivo intentando mejorar este paso se llevaron a cabo distintas pruebas.

Si se sigue solo el camino que inicialmente tenga mayor *der* el costo computacional se reduce en *d* veces, pero al mismo tiempo los resultados son peores respecto a tomar todos los camino y elegir el mejor resultado entre ellos. Para obtener un algoritmo de igual costo computacional que RM y que use solo el camino de mayor *der*, se deben agregar *d*-1 conjuntos de descriptores de partida distintos, lo cual podría dar mejores resultados.

Para verificar esto se llevaron a cabo pruebas usando las bases de datos antes mencionadas tomando nuevamente *d*=7, donde se compararon RM y la alternativa con *d* conjuntos de descriptores de partida siguiendo solo el camino con mayor *der* (esto es así porque sería el descriptor con más probabilidades de mejoría). Los conjuntos de partida se

eligieron aleatoriamente, se tomo la precaución que el conjunto de RM coincida con uno de los *d* conjuntos del nuevo algoritmo, que se llamó RMfs (RM first step). Las comparaciones se realizaron para 100 casos distintos con 100 conjuntos de partida para RM y otras 600 adicionales para RMfs.

Los resultados se volcaron en la Tabla 4.4.1, donde se puede ver que RMfs da mejores resultados (menor *S*) en una mayor cantidad de veces para todas las bases de datos.

Además se agregaron los resultados usando seis y cinco conjuntos de partida en RMfs. Se puede ver que para el caso de seis conjuntos de partida en RMfs los resultados son aún mejores que RM para las cuatro bases de datos. En el caso de cinco conjuntos iniciales en RMfs los resultados son comparables, dando mejores resultados RMfs en dos de las cuatro bases de datos, el total continua siendo favorable a RMfs. Esto indica que RMfs es un algoritmo algo más eficiente que RM.

Existe la posibilidad de que la mejora en RMfs sea debido a que se utilizan distintos conjuntos de partida explorando distintos puntos en el campo de soluciones y no por elegir el camino de mayor *der*, por lo que adicionalmente se comparó RMfs con un algoritmo que también usa solo un camino pero el mismo es elegido de manera aleatoria, se lo denominó RMfsA (RM first step Arbitrary). Se usaron los 700 conjuntos de partida anteriores, los resultados se muestran en la Tabla 4.4.2. Se puede ver que si bien RMfs da mejores resultados indicando que es mejor elegir el camino de mayor *der*, la diferencia no es muy grande indicando también que parte de la mejora en RMfs respecto a RM sería por utilizar distintos conjuntos de partida lo que es equivalente a explorar el espacio de soluciones en distintos sectores al mismo tiempo. Esto último se asemeja a la metodología usada en Algoritmos Genéticos; este punto se desarrollará con mayor detalle en la sección titulada RM y AG Combinados.

Tabla 4.4.1 Número de casos en que los resultados son mejores (menor *S*) comparando los algoritmos RMfs vs. RM para 100 casos distintos usando las cuatro bases de datos

Algoritmo	MES	GI	Fluor	GABA	Total
Nro. de conjuntos de descriptores iniciales RMfs =7					
RMfs	52	57	58	58	225
RM	39	30	36	30	135
Igual	9	13	6	12	40
Nro. de conjuntos de descriptores iniciales RMfs =6					
RMfs	49	53	52	56	210
RM	40	34	40	33	147
Igual	11	13	8	11	43
Nro. de conjuntos de descriptores iniciales RMfs =5					
RMfs	41	49	44	49	183
RM	48	38	46	39	171
Igual	11	13	10	12	46

Tabla 4.4.2 Número de casos en que los resultados son mejores (menor *S*) en la comparación de los algoritmos RMfs vs. RMfsA, usando 700 casos distintos usando las cuatro bases de datos.

Algoritmo	MES	GI	Fluor	GABA	Total
RMfsA	220	218	250	226	914
RMfs	239	237	246	246	968
Igual	240	244	203	227	914

4.5 RM y AG Combinados

Otra alternativa posible para mejorar RM era combinándolo de alguna forma con un algoritmo genético, para poder hacer esto fue necesario llevar a cabo la adaptación de un algoritmo disponible y de esta forma se obtuvo un AG aplicable a estudios QSAR/QSPR. El algoritmo usado como base fue el "*GA Toolbox for MATLAB*" desarrollado en el departamento de Control Automático e Ingeniería de sistemas de la Universidad de Sheffield, UK.[90] Luego de numerosas modificaciones y pruebas se logró ponerlo en funcionamiento y aplicarlo a problemas de QSAR. Los resultados encontrados con estos algoritmos son muy buenos; sin embargo los obtenidos con ERM suelen ser mejores[91-93], y además están presentes las dificultades en la aplicación de AG antes mencionadas

(sección 4.2.4). Estas fueron las razones que llevaron a pensar que al fusionar AG con RM se iba a obtener un nuevo algoritmo que sería superior a cada uno por separado.

Debido a que los AG son muy útiles como punto de comparación y pueden ser un buen punto de partida para futuros trabajos de desarrollo de nuevos algoritmos, se incluyó el código de los AG modificados para QSAR/QSPR en el Apéndice.

4.5.1 Ajuste de parámetros de AG

En las pruebas que se llevaron a cabo se usó el conjunto de datos de fluorofilicidad (FLUOR) antes mencionado, que contiene 116 compuestos orgánicos caracterizados por 1268 descriptores teóricos [80].

Los algoritmos genéticos tienen diversos pasos en su desarrollo que requieren establecer algunos parámetros de ajuste. Tal como se mencionó en la sección 4.2.4 estos no tienen una dependencia lineal por lo que optimizar los mismos es una tarea ardua y no sistemática. Para poder visualizar la forma en que se ajustan los algoritmos genéticos se mostrarán distintos gráficos de *S* en función del número de generación intentando mostrar el comportamiento de los algoritmos genéticos al variar los parámetros.

La optimización de estos parámetros se llevó a cabo probando distintos valores de cada uno, manteniendo el resto constante y luego usando combinaciones de los parámetros que mostraron mejores resultados inicialmente.

Para poder visualizar el desempeño de los AG usando distintos parámetros de ajuste se hicieron gráficos de *S* en función del número de generaciones; el primer grafico muestra el desarrollo de el algoritmo cuando se usan los parámetros estándar:[90] Número de individuos (IND)=20, Brecha generacional (GGAP)=0.9, Probabilidad de entrecruzamiento (CrossP)=0.9 y probabilidad de mutación (MutP)=0.7/*d*. El resto de los gráficos fueron realizados usando distintas variaciones de los parámetros de ajuste, estos están indicados en el texto al pie de las figuras; los parámetros que cambian respecto al estándar están indicados en negrita.

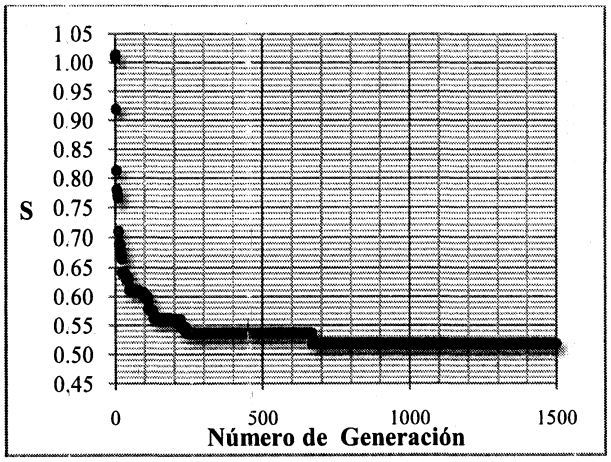

Figura 4.5.1 Desempeño de AG con IND=20, GGAP= 0.9, CrossP=0.9, MutP=0.7/*d*

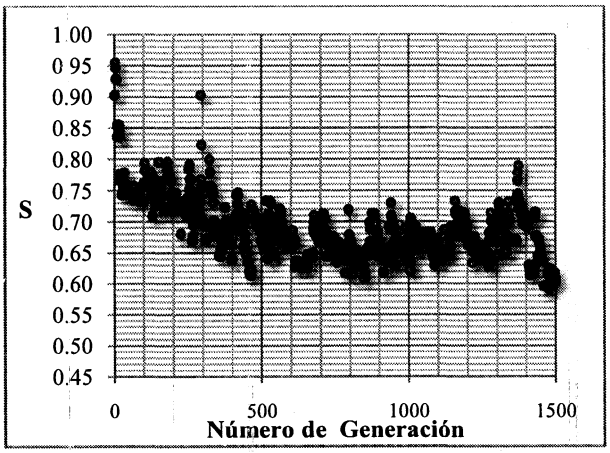

Figura 4.5.2 Desempeño de AG con IND=**5**, GGAP= 0.9, CrossP=0.9, MutP=0.7/*d*

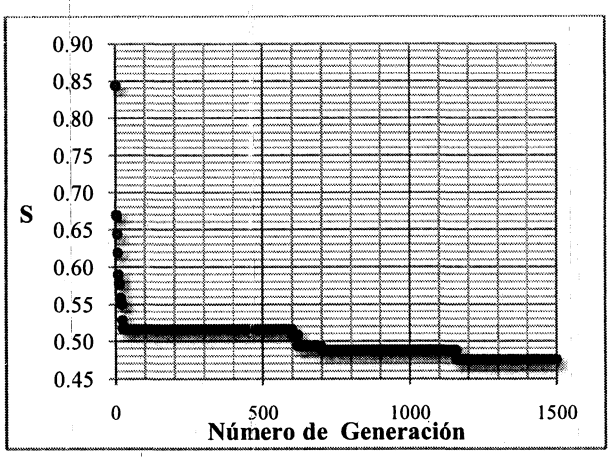

Figura 4.5.3 Desempeño de AG con IND=**100**, GGAP=0.9, CrossP=0.9, MutP=0.7/*d*

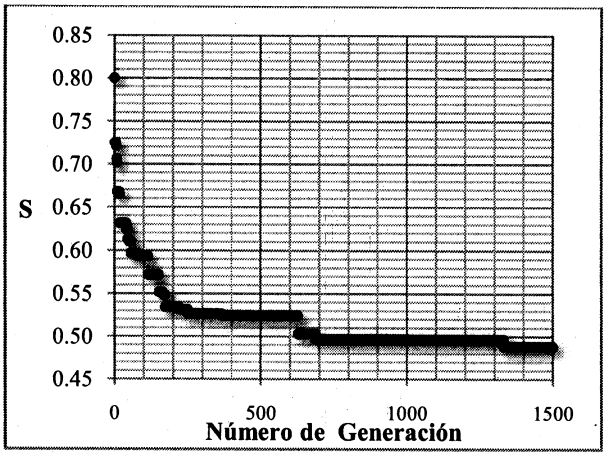

Figura 4.5.4 Desempeño de AG con IND=20, GGAP=**0.5**, CrossP=0.9, MutP=0.7/*d*

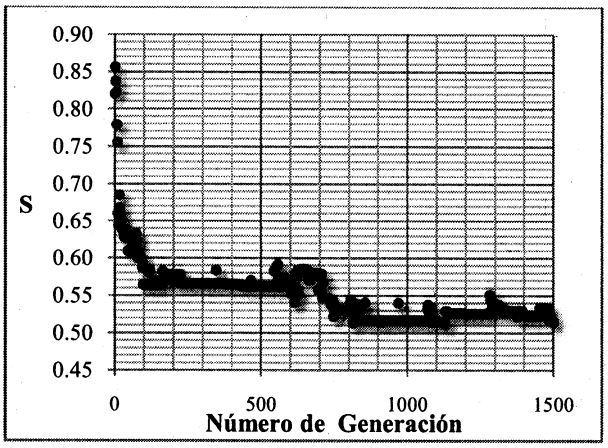

Figura 4.5.5 Desempeño de AG con IND=20, GGAP=**1.25**, CrossP=0.9, MutP=0.7/*d*

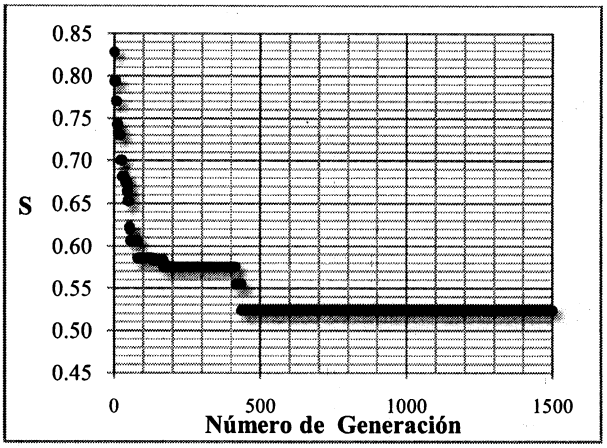

Figura 4.5.6 Desempeño de AG con IND=20, GGAP=0.9, CrossP=**0.2**, MutP=0.7/*d*

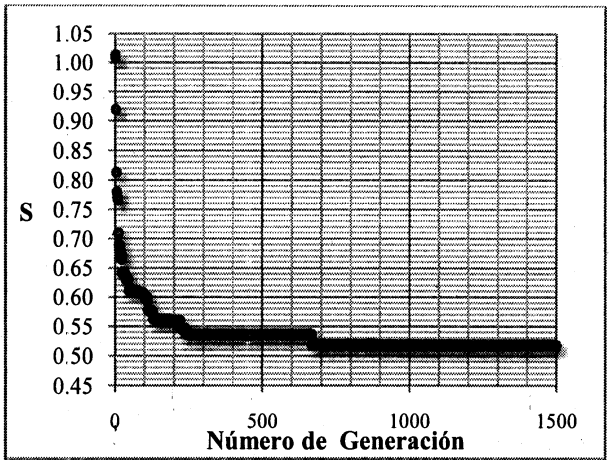

Figura 4.5.7 Desempeño de AG con IND=20, GGAP=0.9, CrossP=**0.9**, MutP=0.7/*d*

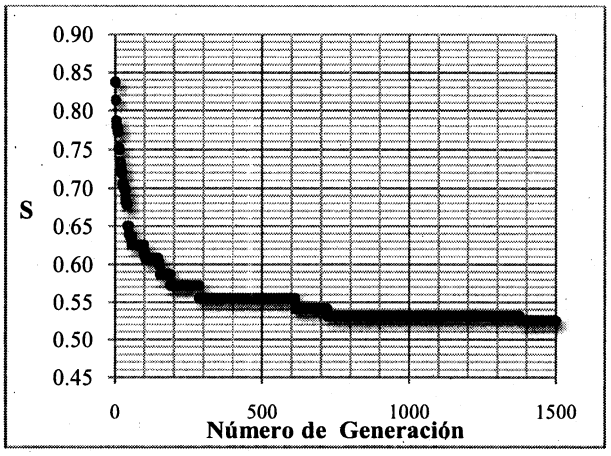

Figura 4.5.8 Desempeño de AG con IND=20, GGAP=0.9, CrossP=0.9, MutP=**0.2**/*d*

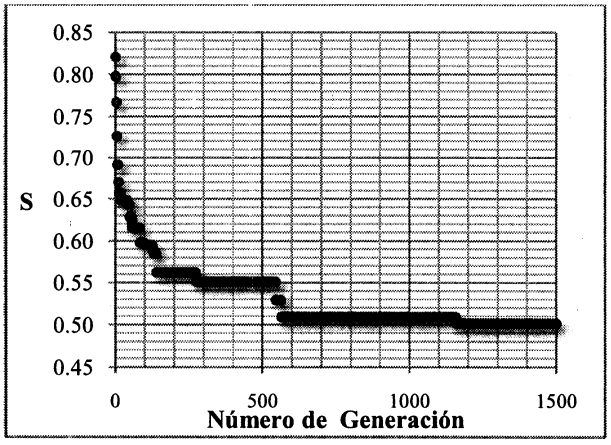

Figura 4.5.9 Desempeño de AG con IND=20, GGAP=0.9, CrossP=0.9, MutP=**1.4**/*d*

Es importante tener en cuenta que estos gráficos solo representan una corrida, por lo que se analizaron varios de cada uno para verificar que el comportamiento entre corridas fuera similar. Para comparar el desempeño en términos de minimización de S se deberán usar los resultados de varias corridas; esto se debe a que los AG son un método estocástico donde los resultados para los mismos parámetros entre dos corridas pueden diferir ya que la población inicial de individuos queda determinada al azar. Por esta razón se llevaron a cabo 100 corridas para cada disposición de parámetros de ajustes, los resultados se pueden ver en la Tabla 4.5.1. En todos los casos el criterio de convergencia elegido fue cuando un individuo ocupó el 90% de la población o cuando el número de generaciones alcanzó 1500.

Tabla 4.5.1 Resultados de AG para distintos parámetros de ajuste luego de 100 corridas.

IND	**20**	**5**	**100**	20	20	20	20	20	20
GGAP	**0.9**	0.9	0.9	**0.5**	**1.25**	0.9	0.9	0.9	0.9
CrossP	**0.6**	0.6	0.6	0.6	0.6	**0.2**	**0.9**	0.6	0.6
MutP	**0.7**	0.7	0.7	0.7	0.7	0.7	0.7	**0.2**	**1.4**
S mínimo	**0.4600**	0.5245	**0.4421**	0.4731	0.4567	0.4575	0.4578	0.4656	0.4546
S promedio	**0.5203**	0.5747	**0.5026**	0.5311	0.5108	0.5226	0.5233	0.5347	0.5269

Al analizar los resultados de la Tabla 4.5.1 no se puede llegar a conclusiones claras salvo para el caso de la variación del número de individuos donde visiblemente al usar 100 individuos mejoran los resultados, siendo estos los mejores resultados encontrados entre todas las distintas pruebas. Adicionalmente se puede apreciar que al aumentar el GGAP mejoran los resultados respecto al estándar (primera fila Tabla 4.5.1 e indicados todos en negrita) de la todos los parámetros en negrita) pero no son mejores que los encontrados para el uso de 100 individuos.

Las pruebas preliminares mostraron que ni la adición de más de 100 individuos ni el cambio de más de un parámetro a la vez presentaron resultados favorables por lo que fueron descartadas. El hecho de que la adición o remoción de individuos tenga un efecto marcado en los resultados obtenidos coincide con el cambio que se ve en los gráficos. Al disminuir el número de individuos el algoritmo muestra un mayor ruido, situación que en algunos casos puede ser favorable para evitar mínimos locales, como podría ser el caso del aumento de GGAP. Sin embargo al aumentar el número de individuos si bien el algoritmo

muestra tener menos variaciones, la minimización de *S* es mucho más eficiente alcanzando menores valores en un número menor de generaciones. Al cambiar el resto de los parámetros las alteraciones en el desempeño de los algoritmos que se pueden observar en los gráficos son mucho menos evidentes.

4.5.2 RM con población inicial

Se llevaron a cabo diversas pruebas para combinar RM y AG, solo una de ellas dio resultados positivos y será presentada en este capitulo. El listado y detalle del resto de las pruebas que no tuvieron éxito se puede ver en la sección 8.3 del Apéndice.

La manera en que se encaró esta combinación de algoritmos fue partir de un algoritmo RM y luego se le agregó un aspecto de AG. Para esto se agregó a RM un conjunto de puntos de partida elegidos al azar de igual forma que en AG; cada punto de partida fue conformado por un grupo *d* de descriptores equivalentes a un individuo en AG y el equivalente a una población en AG fue constituido por un conjunto de dichos puntos de partida.

Tal como se pudo observar en la sección 4.4 es conveniente usar un solo conjunto de descriptores inicial por paso ya que de esta forma se obtienen mejores resultados. Por lo tanto quedaba por resolverse el número óptimo de individuos a incluir en la población.

Se llevaron a cabo pruebas para comparar el nuevo algoritmo combinado con RM y AG que se denominó RM_p. Se usó nuevamente el conjunto de datos de fluorofilicidad (FLUOR) antes mencionado, que contiene 116 compuestos orgánicos caracterizados por 1268 descriptores teóricos [80], d=7 y 100 corridas. Los resultados se volcaron en la Tabla 4.5.2. Para el caso del nuevo algoritmo el número de individuos usado es igual a 7 para que de esta forma los resultados sean comparables con RM (ver sección 4.4)

Tabla 4.5.2 Comparación de AG, RM y RM_p el cual consta de una población inicial aleatoria

Algoritmo	*S* mínimo	*S* promedio
AG	0.4421	0.5026
RM_p	**0.4342**	**0.4870**
RM	0.4408	0.4922

En la tabla se puede observar que los mejores resultados, tanto en el mínimo encontrado entre todas las soluciones como en el promedio de las mismas, corresponden a RM_p siendo este un algoritmo combinado que mejora a cada algoritmo por separado.

La forma en que se desarrolla el nuevo algoritmo difiere de AG, en este caso cada individuo es calculado por separado y luego se selecciona aquel que presenta mejores resultados. Por lo tanto al agregar más individuos la solución necesariamente mejora o permanece igual, es decir al aumentar el número de individuos los resultados siempre tenderán a mejorar. Entonces quedaría por determinar el número óptimo en términos de costo computacional vs mejoría de los resultados.

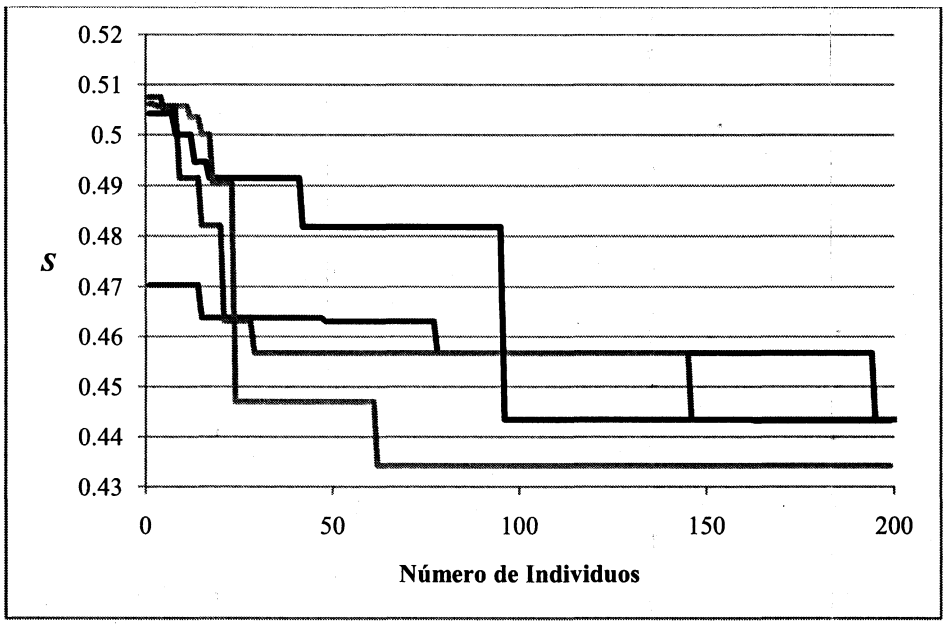

Figura 4.5.10 Disminución de S al aumentar el número de individuos en la población para RM_p

Los resultados muestran que no existe un único número de individuos que optimice el costo computacional en función de los resultados lo que es razonable por tratarse de un proceso aleatorio. Como se mencionara, siempre que se pueda es conveniente agregar más individuos, sin embargo esto acarrea un aumento del esfuerzo computacional.

Del gráfico podría decirse que un número razonable sería 100 individuos, ya que las soluciones son aceptables para todas las pruebas usando tal población.

4.6 *Pruebas para mejorar ERM*

Tal como se mencionó en la sección 4.3.1 el algoritmo ERM tienen una menor sensibilidad frente a cambios en el conjunto inicial de descriptores en comparación a RM, sin embargo si bien en menor medida la solución final alcanzada depende de esta elección inicial. Por lo tanto sería conveniente encontrar una forma de determinar que conjunto de descriptores inicial usar y/o comprobar si lo encontrado para RM en la sección 4.4 es aplicable a ERM.

Para esto se llevaron a cabo diversas pruebas equivalentes a las hechas para el caso de RM, estas fueron expuestas a continuación.

Adicionalmente se realizaron pruebas con un algoritmo que tomaba un S muy alto como punto de partida. Estas fueron motivadas suponiendo que al comenzar el algoritmo ERM lo más lejos posible de algún mínimo local de S se tendrían menos posibilidades de permanecer en el mismo. Dada la complejidad de este algoritmo, los resultados obtenidos no justificaron su inclusión en este capítulo, sin embargo sus detalles pueden encontrarse en la sección 8.4 del Apéndice.

4.6.1 Primer paso de ERM

De igual forma que en el caso de RM se llevaron a cabo pruebas para comprobar si sería conveniente usar un algoritmo ERM que busque a partir de un único conjunto de descriptores todos los caminos posibles o uno que busque solo el camino con mayor *der* por conjunto, completando con otros conjuntos aleatorias para igualar el esfuerzo computacional. Este nuevo algoritmo se denominará ERM_{fs} (ERM first step). Las pruebas al igual que en la sección 4.4 se realizaron con d=7, para 100 casos distintos usando 100 conjuntos de partida para ERM y otras 600 aleatorias adicionales para ERM_{fs}. Solo se usó la base de datos denominada FLUOR[80].

Los resultados se volcaron en la Tabla 4.6.1, donde además se agregaron los resultados para el caso de usar seis, cinco y cuatro conjuntos de partida en RM_{fs}. Se puede observar que ERM_{fs} da mejores resultados (menor S) que ERM para el caso de igual esfuerzo computacional (siete conjuntos en ERM_{fs}) y Para los casos de menor esfuerzo computacional usando seis y cinco conjuntos de partida en ERM_{fs}. Recién cuando se disminuye el número de conjuntos de partida a cuatro en ERM_{fs} los resultados se invierten

siendo favorables a ERM. Esto indicaría que ERM_{fs} es un algoritmo aún más eficiente que que ERM.

Tabla 4.6.1 Número de casos en que los resultados son mejores (menor *S*) comparando los algoritmos ERM_{fs} vs. ERM para 100 casos distintos para la base de datos FLUOR.

Algoritmo	*Número de soluciones iniciales* ERM_{fs}			
	7	6	5	4
ERM_{fs}	51	46	42	34
ERM	27	34	37	42
Igual	22	20	21	24

4.6.2 ERM con población inicial

Anteriormente se ha visto que ERM es un algoritmo que supera a RM. Por otro lado se ha logrado mejorar RM combinándolo con AG obteniendo un algoritmo RM con población de partida aleatoria (RM_p). Emerge de esto la posibilidad de combinar ERM con AG y obtener un algoritmo que sea superior aún a ERM. Cabe mencionar que en la aplicación de ERM se había probado su superioridad frente a AG. (ver secciones 6.3, 6.4 y 6.5) Las pruebas se llevaron a cabo en las mismas condiciones que en la sección 4.5.2, es decir se usó nuevamente el conjunto de datos de fluorofilicidad (FLUOR) [80], d=7 y un número de corridas igual a 100. Los resultados se volcaron en la Tabla 4.6.2. Nuevamente para el caso del nuevo algoritmo el número de individuos usado es igual a 7 para que de esta forma los resultados sean comparables con ERM.

Tabla 4.6.2 Comparación entre AG, ERM y ERM_p el cual consta de una población inicial aleatoria

Algoritmo	*S* mínimo	*S* promedio
AG	0.4421	0.5026
ERM_p	**0.4328**	**0.4423**
ERM	**0.4328**	0.4477

En la tabla se puede observar que los mejores resultados corresponden a ERM_p siendo este un algoritmo combinado que mejora a cada algoritmo por separado. Este es por lo tanto el mejor algoritmo encontrado hasta el momento.

Hay que notar que si el número de individuos aumenta los resultados tenderían a mejorar, como antes se mencionó.

Quedaría por determinar el número óptimo de individuos a utilizar en términos de costo computacional; para esto se llevaron a cabo pruebas adicionales agregando individuos de a uno.

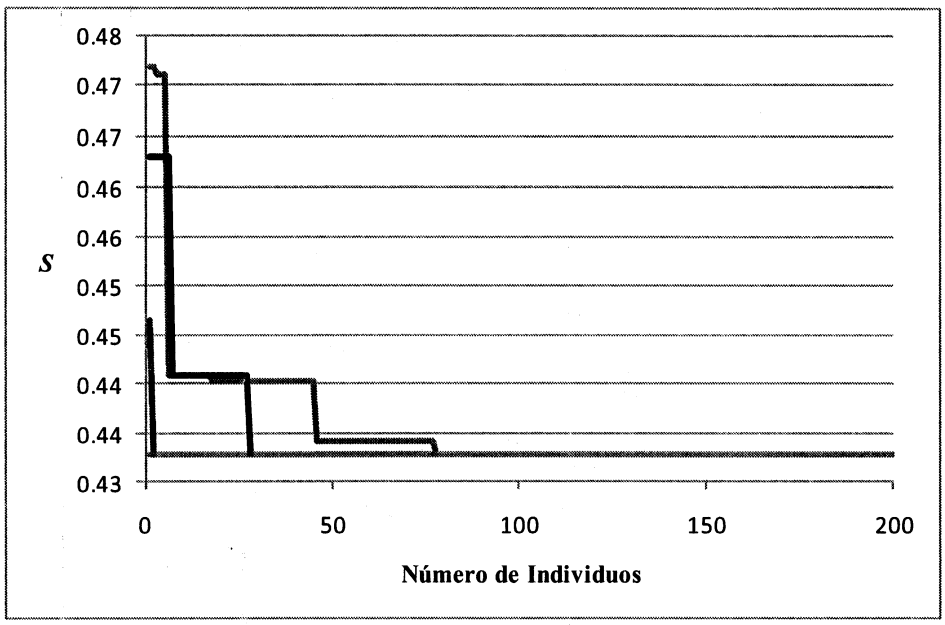

Figura 4.6.1 Disminución de *S* al aumentar el número de individuos en la población para ERM_p

Nuevamente los resultados muestran que no existe un número óptimo exacto de individuos por ser un proceso aleatorio. También se observa que siempre que sea posible es conveniente agregar más individuos para obtener mejores resultados, sin olvidar que esto trae acarreado un aumento del esfuerzo computacional.

A diferencia de lo observado en el grafico similar para RM_p la desviación estándar disminuye con mayor velocidad por lo que en este caso un número razonable sería 50 individuos. Cabe mencionar que es recomendable reducir el número de individuos cuando se realicen pruebas preliminares durante cualquier estudio en que se use el algoritmo para reducir el tiempo de cálculo.

4.7 *Determinación del número optimo de descriptores a incluir en un modelo*

Un paso fundamental in la construcción de un modelo QSPR/QSAR es determinar el número optimo de descriptores (d_{opt}) a ser incluidos en el modelo. Esto no suele ser fácil ya que a medida que el número de descriptores es aumentado los parámetros estadísticos para el conjunto de calibración (moléculas usadas para ajustar el modelo) siempre tienden a mejorar. Pero por otro lado los parámetros estadísticos del conjunto de validación (moléculas que se han dejado a un lado para probar el poder predictivo del modelo) en un principio mejoran, al ir incluyendo más información relevante de la estructura, para luego de un cierto numero de descriptores, que depende del caso estudiado, comienzan a deteriorarse. Este deterioro en general es debido a un sobre-ajuste del modelo a las moléculas del conjunto de calibración. Un esquema que resume el problema de sobre-ajuste se puede ver en la Figura 4.7.1[94]; en este gráfico el número optimo de descriptores sería el que presenta un mínimo en la curva de validación ya que los modelos con esta cantidad de descriptores serán los que tengan mejor poder predictivo.

Se podría pensar que una forma de determinar el número óptimo de descriptores sería desarrollar modelos que vayan incluyendo un mayor número de estos, para luego probarlos en el conjunto de validación y elegir el que tenga mejores parámetros estadísticos. A pesar de que esto pueda parecer apropiado no lo sería ya que el modelo elegido habría sido determinado haciendo uso del conjunto de calibración y consecuentemente este dejaría de ser un verdadero conjunto de moléculas externo.

Por lo tanto es necesario un método para determinar d_{opt} que solo use el conjunto de calibración. Para llevar a cabo esto en nuestro grupo de investigación se usaba[73, 91] la función de Kubinyi (*FIT*)[95, 96] la cual es un parámetro estadístico que esta relacionado con el de Fisher (*F*), pero evita la principal desventaja de este ultimo que es muy sensible a cambios en valores bajos de d y poco sensible a los cambios cuanto d tiene valores altos. *FIT*(**d**) tiene baja sensibilidad a cambios en d bajos y una sensibilidad que se incrementa sustancialmente para valores altos de d. Mientras mayor es el valor de *FIT* mejor será el modelo encontrado, su expresión es:

$$FIT = \frac{R^2(N-d-1)}{(N+d^2)(1-R^2)} \qquad (4.6.1)$$

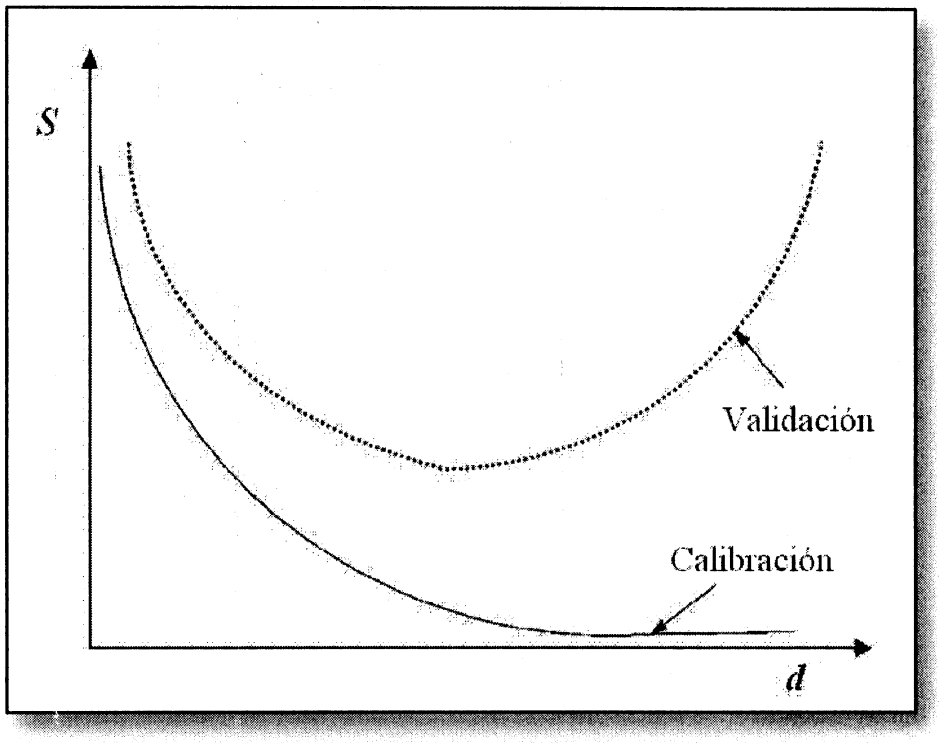

Figura 4.7.1 Comportamiento típico de la desviación estándar de un modelo en los conjuntos de calibración y validación a medida que aumenta *d*.

Se espera que un gráfico de *FIT* vs. d posea un máximo (d_{max}) del que se pueda calcular el número optimo de descriptores (d_{opt}) a incluir en el modelo lineal usando el siguiente criterio:

- si $d_{max} < 7$, entonces $d_{opt} = d_{max}$
- si $d_{max} > 7$, se define $d_1 = \left[\frac{d_{\max}}{2}\right] + 1$, donde $[x]$ denota la parte entera de x. Luego si la pendiente de FIT en d_1 es mayor que en d_1+1, entonces $d_{opt} = d_1$, de otra forma $d_{opt} = d_1+1$

De esta forma el valor de d_{opt} obtenido reflejaría un punto límite a partir del cual la mejora en FIT podría considerarse insignificante.

Sin embargo hay numerosas ocasiones en las que este máximo no se alcanza luego de agregar un número razonable de descriptores al modelo. Por este motivo recientemente hemos propuesto una función *FIT* variable que hemos llamado *VFIT* el cual incluye una

constante semi-empírica k la cual otorga mayor peso al número de descriptores d en la ecuación *FIT*.[92, 93] La nueva expresión tiene la forma:

$$VFIT = \frac{R^2(N - kd - 1)}{(N + d^2)(1 - R^2)} \qquad (4.6.2)$$

Usando esta ecuación se pude obtener d_{opt} como el número de descriptores (d) que de a lugar un modelo con mayor valor de *VFIT* (d_{max}), el mismo se podría ver en un gráfico de *VFIT* vs. d. Para poder usar *VFIT* previamente hay que determinar la constante k, esto se hace tomando valores incrementales de k en 0.5 hasta que el valor máximo en *VFIT* permanezca inalterado al menos durante dos incrementos[97] y cumpla con la regla que por lo menos deben existir 5 datos experimentales por cada parámetro de calibración usado.[98]

De esta forma se obtiene d_{opt} sin usar en ningún momento los datos del conjunto de validación. Ejemplos de aplicación del método pueden verse en las secciones 6.4 y 6.5. Este método claramente puede ser usado con ERM_p, para esto es recomendable usar durante las pruebas una población no muy grande. Luego cuando se haya determinado d_{opt}, realizar una nueva búsqueda con un número alto de individuos en la población de partida solamente para ese número optimo de descriptores. De esta forma se obtendría el mejor modelo posible con un costo computacional no muy elevado.

Un punto a tener en cuenta para futuros estudios es que si el número de datos experimentales es lo suficientemente grande se podría investigar la posibilidad de dividir el mismo en un conjunto de calibración y dos conjuntos de validación distintos. Con uno de ellos se elegiría el mejor modelo y el segundo conjunto sería el verdadero conjunto externo que no fue usado en ningún momento para determinar el modelo y por ende pude ser usado para terminar el verdadero poder predictivo del modelo encontrado.

4.8 Conclusiones

Se mostró un resumen de los métodos de búsqueda que optimizan la selección de un conjunto de descriptores entre un conjunto mucho más grande de los mismos.

Se expuso un nuevo algoritmo denominado ERM que está basado en RM y presenta similitudes con la metodología de Simulated Annealing; los resultados de ERM son mejores que los de RM.

Por otro lado se investigó la posibilidad de combinar los algoritmos genéticos con RM para obtener un algoritmo que sea superior a ambos por separado. De todas las opciones probadas la única que cumplió con esto fue la de un algoritmo RM con una población inicial aleatoria.

Asimismo se probó la combinación de algoritmos genéticos con ERM encontrando que el nuevo algoritmo ERM_p supera incluso a ERM y se presenta como el mejor algoritmo.

Por ultimo se mostró un nuevo método para la determinación del número óptimo de parámetros que se deben incluir en un modelo haciendo uso de ERM, pudiendo usarse también ERM_p de la misma forma.

5 Validación

"Nunca pienso en el futuro. Llega enseguida" (Albert Einstein)

5.1 Introducción

Cuando se ajusta cualquier tipo de modelo predictivo a un conjunto de datos, es esencial verificar que el modelo pueda generalizarse a futuros datos del mismo tipo. Esto es particularmente importante cuando se ajustan modelos complejos. [36]

Por lo tanto este paso es crítico en el diseño de un modelo, este paso determinará si el modelo es capaz de predecir datos no usados en la calibración, en la cual como ya se ha mencionado se encuentra y ajusta el modelo óptimo.

Pude suceder que uno tenga una muy buena calidad en la calibración y una pésima calidad en la validación. Básicamente esto pude deberse a un sobre-ajuste como fuera mencionado en la sección 4.7 o a una correlación fortuita[99], lo que significa que hubo una coincidencia casual entre la propiedad y la estructura molecular; claramente si aparece este tipo de correlación el modelo no servirá para predecir datos externos al conjunto de calibración.

Se espera que los errores obtenidos en la validación sean comparables a los encontrados previamente en el conjunto de calibración. En la literatura se pueden encontrar diferentes estrategias de validación del modelo para estimar su habilidad predictiva y también para elucidar posibles correlaciones fortuitas. En general, la validación se lleva a cabo de las dos maneras siguientes: fraccionando el conjunto total de N moléculas, separando un conjunto de validación, o se usa el propio conjunto de calibración para validar el modelo en forma teórica. Ambas metodologías describen las características predictivas del modelo usando compuestos para la validación que no estuvieron involucrados durante el ajuste de los datos, sin embargo para la validación cruzada los compuestos fueron usados en la selección de los descriptores del modelo.

5.2 *Distribución en conjuntos de calibración y validación*

Este procedimiento se ha aplicado desde hace 60 años en QSPR-QSAR[100]. La distribución de N moléculas con datos conocidos de la propiedad en un conjunto de calibración (c) y en uno de validación (v), con c y v compuestos respectivamente, permite efectuar una "validación externa del modelo". En la misma se emplea el modelo encontrado durante la calibración para predecir los valores de la propiedad de las moléculas del conjunto de validación. Luego, se compara la propiedad experimental contra la predicha y se obtienen los parámetros estadísticos de validación R_{val} y S_{val}, que miden la calidad de los modelos encontrados.

Se debe notar que los parámetros estadísticos R y S de calibración y de validación no deben tener una diferencia muy grande para que se pueda considerar que el modelo posee capacidad predictiva.

Realizar una validación externa del modelo solo es factible cuando se disponen de un número grande de compuestos con valores experimentales conocidos. Cabe mencionar que en los trabajos de los últimos años en general resulta poco atractivo y poco común emplear un conjunto de validación con muchos compuestos, ya que usualmente es poco práctico y a veces hasta imposible. [36]

5.3 *Validación interna o teórica*

Se denomina validación interna o teórica al método por el cual es posible analizar el poder predictivo de un modelo usando solamente los datos del conjunto de calibración. Estos estudios son más recientes y dependen fuertemente de los recursos computacionales disponibles. Los ejemplos más simples de validación teórica del modelo son los estudios la técnica de validación cruzada (VC) y de la variable **Y** aleatoria.

5.3.1 Validación Cruzada

La técnica de validación cruzada (VC) [36, 101-103] es la que se usa más comúnmente en la literatura QSPR-QSAR para analizar el poder predictivo de un

modelo. Es posible implementar el método con diferentes variantes tales como dejar-uno-afuera (*loo*, del ingles *leave-one-out*) y dejar-n%-afuera (*l-n%-o*), siendo *n%* el porcentaje de moléculas que se remueven del conjunto por etapas y corresponde a $n.N \times 10^{-2}$ moléculas. Para practicar el método *l-n%-o* sobre **c**, se debe primero decidir el grado de predictibilidad que deba superar el modelo, representado numéricamente por el porcentaje de moléculas extraídas.

Supongamos que a partir de las N moléculas se arman subconjuntos aleatorios $\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, \ldots, \mathbf{c_x}$ conteniendo $n.N \times 10^{-2}$ moléculas cada uno, en los cuales sus elementos no se repiten ni en el mismo ni en subconjuntos diferentes. Entonces el procedimiento *l-n%-o* sería se la siguiente forma:

1) Se remueve el conjunto $\mathbf{c_1}$ de **c** y se calibra nuevamente el modelo con las restantes $N - n.N \times 10^{-2}$ moléculas para predecir con éste la propiedad de las moléculas extraídas.

2) Se devuelven las moléculas de $\mathbf{c_1}$ a **c** y se remueve otro conjunto diferente $\mathbf{c_2}$ y se vuelve a calibrar el modelo con las $N - n.N \times 10^{-2}$ moléculas restantes, y nuevamente se predice la propiedad en el conjunto $\mathbf{c_2}$.

3) Se continúa con este procedimiento hasta predecir la propiedad para las N moléculas. Ahora se hace una correlación entre los valores predichos para las N moléculas y los experimentales y se obtienen los parámetros estadísticos de validación cruzada, $R_{l\text{-}N\%\text{-}o}$ y $S_{l\text{-}n\%\text{-}o}$ asociados al arreglo de moléculas $\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, \ldots, \mathbf{c_x}$

4) Repetir todo el procedimiento a partir de 1) para todos los arreglos posibles de moléculas $\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, \ldots, \mathbf{c_x}$

Es conocido que valores bajos de $R_{l\text{-}N\%\text{-}o}$ conducen a modelos de bajo poder predictivo, pero la situación inversa no necesariamente es cierta.[104] La manera más sencilla de hacer *VC* es remover sucesivamente del conjunto **c** una molécula a la vez lo que sería un procedimiento *loo*. En el caso de *l-n%-o*, el costo computacional se incrementa considerablemente al convertirse el problema en uno de combinatorias consistente en elegir arreglos de subconjuntos de $n.N \times 10^{-2}$ moléculas elegidas entre N. Debido a que el número de combinatorias es enorme, es bastante común ensayar la validación cruzada con un determinado número de casos de arreglos generados al azar. Luego el análisis de los parámetros estadísticos encontrados se hará para el peor caso encontrado es decir para el de mayor S y menor R, ya que se está intentando determinar si el modelo falla en la predicción para alguna selección de moléculas.

En una distribución de N moléculas se cumple la condición de conservación:

$$N = c + v \tag{5.1.1}$$

El hecho de que cualquier método estadístico funcione mejor cuanto mayor sea el número de datos analizados es algo conocido en cualquier técnica de análisis de datos, esto es debido a que los parámetros calculados a partir de los mismos se tornan más representativos de la población. Este simple requerimiento indica que las cantidades **c** y **v** deben ser lo más grande posible. Aparte, y como es usual en estadística, existe también un retorno de escala disminuyente de forma que si **c** o **v** son muy grandes (varios cientos de moléculas) será de poco beneficio hacerlos aún mayores. La maximización de los subconjuntos hará que exista una situación de compromiso en el número de moléculas que posean **c** y **v**, ya que se debe cumplir con (5.1.1).

En aquellas circunstancias en que N sea grande será posible aplicar una partición molecular sin inconvenientes, pero para los casos que traten con pocas moléculas será mejor recurrir a *VC* como técnica de validación. Esto es consecuencia de que el método *l-n%-o* maximiza el tamaño de **c** y **v**, pues la calibración se efectúa con todas las moléculas y lo mismo ocurre en la validación.

5.3.2 Variable Y aleatoria

Se trata de una técnica que se usa frecuentemente para analizar el poder predictivo del modelo. En esta prueba, se le asigna valores aleatorios a la propiedad experimental **P** (variable dependiente **Y**) y se usa la misma matriz de descriptores independientes para calcular nuevamente un modelo. El proceso se repite un gran número de veces. Es de esperar que los modelos QSPR resultantes tengan valores bajos de R y R_{val}. Si no sucede esto y aparecen casos de modelos que poseen buena calidad, entonces son indicadores de la existencia de correlación fortuita y de redundancia de descriptores en el modelo.[105] Si todos los modelos QSPR obtenidos de la **Y** aleatoria tienen valores altos de los parámetros R y R_{val} implicaría que no se podrá obtener un modelo QSPR con el método de modelado usado.

5.3.3 Nuevo método de validación propuesto

Cómo ya se mencionara anteriormente, cuando el número de moléculas disponibles es alto, mayor a 100, [36] no existe problema en dividir el conjunto total en uno de calibración y otro de validación. Sin embargo cuando el número de moléculas es menor, esto no es posible sin perder calidad tanto en la calibración como en la validación. Por lo tanto normalmente en esos casos se recurre a una validación cruzada, sin embargo esta no certifica el poder predictivo de un modelo ya que los parámetros de validación cruzada sean aceptables es condición necesaria pero no suficiente para determinar el poder predictivo de un modelo.[104]

Por este motivo se desarrolló e implementó en distintas aplicaciones un método nuevo el cual agrega un paso adicional de validación a las validaciones teóricas cruzadas y de Y aleatoria.[92, 93]

El mismo consta de los siguientes pasos:

- Se lleva a cabo un estudio QSAR-QSPR de forma habitual buscando el mejor modelo entre el conjunto total de descriptores usando un conjunto de calibración maximizado. A este modelo inicial lo llamaremos m_i.
- Se valida m_i en forma teórica, calculando R_{loo}, $R_{l\text{-}n\%\text{-}o}$, S_{loo} y $S_{l\text{-}n\%\text{-}o}$. corroborando que estos parámetros hallados sean satisfactorios[104]
- Luego se procede a remover un número determinado de moléculas del conjunto de calibración tomando un conjunto de validación.
- Se vuelve a calibrar el modelo (m_i) con el nuevo conjunto de calibración reducido y se comprueba que los parámetros R_{val} y S_{val} sean aceptables (similares a los parámetros de calibración). Cabe mencionar que este paso no sería valido ya que las moléculas del conjunto de validación habían sido usadas para la selección del modelo inicial (m_i) y por lo tanto no pueden ser consideradas como realmente externas.
- Para resolver este problema se vuelve a hacer una búsqueda entre todos los descriptores disponible pero ahora usando el nuevo conjunto de calibración, y se encuentra un nuevo modelo (m_j); al que se le calculan R_{val} y S_{val}.
- Por ultimo se compara el nuevo modelo (m_j) y el modelo inicial (m_i) y si estos son similares, tanto en los descriptores encontrados como en los parámetros estadísticos, se pude considerar que la remoción de las moléculas para el conjunto de validación no afectó la selección de descriptores en forma apreciable y por lo tanto el

modelo inicial (m_i) pude ser validado con las moléculas del nuevo conjunto de validación.

Cabe mencionar que luego es recomendable usar en cualquier predicción futura el modelo inicial (m_i) ya que este fue calibrado con más moléculas y por lo tanto posee más cantidad de información estructural.[92, 93]

6 Cálculo de Propiedades: Aplicaciones QSPR-QSAR

"La ciencia más útil es aquella cuyo fruto es el más comunicable" (Leonardo da Vinci)

6.1 Análisis QSPR de la Fluorofilicidad de compuestos orgánicos

6.1.1 Introducción

En la actualidad, la química de compuestos fluorados presenta muchas aplicaciones interesantes en síntesis y catálisis. La tendencia de una sustancia orgánica a disolverse en un medio fluorado ha ganado importancia de manera continua luego del descubrimiento de la catálisis en una bi-fase fluorada en 1994,[106] esto es debido a que las reacciones en una bi-fase tienen la ventaja de que las fases fluorada y orgánica son inmiscibles a temperatura ambiente y se homogenizan al elevar la temperatura. Por lo tanto uno puede exponer un reactivo en la fase orgánica a un catalizador en la fase fluorada simplemente calentando y luego separar los productos (en la fase orgánica) del catalizador (fase fluorada) bajando la temperatura. Otras ventajas útiles de estas técnicas basadas en solventes fluorados son la propiedades físicas y químicas únicas que poseen estos solventes al ser no tóxicos, inertes y de fácil purificación.[107]

La fluorofilicidad de un compuesto se puede cuantificar haciendo uso de la constante de partición (P) entre una fase fluorada $(CF_3C_6F_{11})$ y una orgánica $(CH_3C_6H_5)$.[108]

$$\ln P = \ln\left[\frac{c(CF_3C_6F_{11})}{c(CH_3C_6H_5)}\right] \qquad T = 298\ K \tag{6.1.1}$$

Es ampliamente conocido que el diseño experimental de moléculas fluorofílicas requiere un contenido de flúor de al menos 60%, la presencia de una o más cadenas

alquílicas perfluoradas y la ausencia de enlaces por puente de hidrogeno o grupos polares que puedan interactuar con la fase orgánica. Adicionalmente, la fluorización de moléculas frecuentemente se hace agregándole cadenas largas.[109]

Evidentemente, el diseño de catalizadores de bi-fase fluorada se verá ampliamente mejorado si se pude predecir de manera segura la fluorofilicidad de una sustancia dada. Una manera aceptada de forma general para sobrellevar la falta de datos experimentales en fenómenos químicos complejos es el análisis por QSPR,[3] el cual en el presente caso puede presentar predicciones adecuadas de fluorofilicidad.

Varios estudios QSPR de fluorofilicidad fueron publicados en los últimos 7 años. En 2001, Kiss et al [110] estimaron esta propiedad para 59 moléculas orgánicas fluoradas usando Redes Neuronales (NN) de ocho descriptores moleculares elegidos de entre casi cien variables. En 2002, Huque et al[111] emplearon una modificación de relaciones lineales libres de energía (LFER) en 91 químicos para llegar a un modelo de cinco descriptores con interpretación estructural, caracterizados por parámetros estadísticos $R = 0.9742$ y $S = 0.566$. En 2004, Duchowicz et al [112] empleo el mismo conjunto de datos para proponer un modelo diferente basado en regresiones lineales, usando como descriptores átomos y enlaces químicos.

En 2004, de Wolf et al [113] aplicó un modelo universal de lipofilicidad basado en teoría de soluciones orden/desorden (MOD) para predecir la constante de partición de 88 moléculas en ambos PFMCH/tolueno o FC-72/benceno. Sin embargo esas predicciones requerían el conocimiento de volúmenes moleculares y parámetros no específicos de cohesión modificados para el soluto, datos que normalmente no están disponibles. El mismo año, Daniels et al [114] propuso un LFER modificado para 93 compuestos orgánicos usando cinco descriptores de superficie de área, consiguiendo un $R = 0.9716$ y un $S = 0.638$; mostrando una precisión casi idéntica a los modelos antes publicados.

El presente estudio presenta la predicción de 116 compuestos orgánicos cuyos datos experimentales fueron recolectados de dos trabajos anteriores.[111, 115] Para llevar a cabo este trabajo se usaron dos estrategias basadas en regresiones lineales la Regresión de a Pasos[74] (FSR) y el Método de Reemplazo (RM).[74, 77, 79, 116]

6.1.2 Resultados y Discusión

Se aplicó el RM para buscar la mejor relación estructura-fluorofilicidad encontrando que la ecuación que entrega los mejores resultados de validación cruzada $R_{l\text{-}15\%\text{-}o}$ y $S_{l\text{-}15\%\text{-}o}$ conteniendo siete descriptores moleculares de distintas clases es la siguiente:

$$\begin{aligned} \ln P \;=\; & -5.688(\pm 0.5) - 0.348(\pm 0.01)\, SEigp + 0.164(\pm 0.02)\, RDF055p \\ & + 0.0531(\pm 0.05)\, MAXDP - 0.197(\pm 0.01)\, Har + 1.178(\pm 0.1)\, CIC1 \\ & - 6.488(\pm 0.9)\, MATS1v \;+\; 1.606(\pm 0.1)\, HOMA \end{aligned} \qquad (6.1.2)$$

$N = 116,\ R = 0.9806,\ S = 0.494,\ F = 387.7,\ p < 10^{-4},\ AIC = 0.280,\ FIT = 16.450$

$R_{loo} = 0.9778,\ S_{loo} = 0.511$

$R_{l\text{-}15\%\text{-}o} = 0.9677,\ S_{l\text{-}15\%\text{-}o} = 0.620$

donde los errores absolutos de los coeficientes de la regresión están en paréntesis, R es el coeficiente de correlación del modelo, F es el cociente de Fisher, p la significancia del modelo, AIC el criterio de información de Akaike[117, 118] y FIT es la función de Kubinyi[95, 96]. En la Tabla 6.1.1 se presenta una breve descripción de las variables de los modelos presentados.

Al aplicar FSR no se mejora la calidad de la relación ya que lleva a la siguiente ecuación:

$$\begin{aligned} \ln P \;=\; & -2.238(\pm 0.6) + 0.151(\pm 0.005)\, RTm - 1.902(\pm 0.2)\, HATSp \\ & - 0.00611(\pm 0.0007)\, PCD + 0.526(\pm 0.07)\, H2e - 0.362(\pm 0.06)\, C\text{-}006 \\ & + 0.760(\pm 0.2)\, N\text{-}068 \;+\; 0.807(\pm 0.3)\, GATS1p \end{aligned} \qquad (6.1.3)$$

$N = 116,\ R = 0.9740,\ S = 0.572,\ F = 285.5,\ p < 10^{-4},\ AIC = 0.375,\ FIT = 12.110$

$R_{loo} = 0.9700,\ S_{loo} = 0.5932$

$R_{l\text{-}15\%\text{-}o} = 0.9281,\ S_{l\text{-}15\%\text{-}o} = 0.9153$

La Tabla 6.1.2 muestra los valores de fluorofilicidad experimentales y predichos y los correspondientes residuos entre paréntesis. Se pueden considerar los esteres aromáticos **64** ($R_{f7}C(O)OCH_2Ph$) y **65** (p-$R_{f7}C(O)OCH_2C_6H_4OCF_3$) como *outliers* con un residuo que excede $3S$. No es posible discernir si esta desviación es una consecuencia estadística de la presente selección de descriptores en la Ec. (6.1.2) o un resultado físico significativo. Es posible que estos compuestos sean estructuralmente

diferentes al resto de los compuestos del grupo de calibración. En muchos casos los residuos son menores que los de Huque et al [111] que también fueron incluidos en la Tabla 6.1.2. Cabe mencionar que no fue posible incluir cuatro moléculas (**38**, **39**, **40** y **44**) del trabajo anterior ya que la versión disponible del software de cálculo de descriptores tiene un límite de 100 átomos. Por el otro lado aquellas moléculas identificadas por Huque et al [111] como *outliers* y omitidas en su modelo final (**63**, **64**, **65**, **82**, **93**, **94**, **95**, **96**) fueron incluidas en los cálculos del presente trabajo.

El grafico de valores de fluorofilicidad calculados vs. experimentales mostrado en la Figura 6.1.1 indica que los 116 compuestos siguen una línea recta. La Figura 6.1.2 presenta los residuos en relación con los datos experimentales de fluorofilicidad, y muestra que los mejores descriptores moleculares dados en Ec. (6.1.2) llevan a un modelo que sigue una distribución normal y no presenta patrones indeseables que probablemente indicarían la presencia de factores no-modelados contribuyendo a la fluorofilicidad.

La matriz de correlación para la Ec. (6.1.2) (indicada en la Tabla 6.1.3) revela el hecho que existe un grado de intercorrelación entre los descriptores *SEigp* y *Har* (R_{ij}=0.9738), sin embargo estos descriptores tienen información estructural no solapada que hace que el modelo exhiba capacidad predictiva adecuada en la validación cruzada *l-10%-o*, llevada a cabo para 100000 casos generados aleatoriamente.

La estandarización de los coeficientes de regresión[74] en Ec. (6.1.2) permite asignar mayor importancia a las variables en el modelo que poseen mayor valor absoluto de coeficiente estandarizado (en paréntesis), por lo tanto se obtiene el siguiente ordenamiento de contribuciones a ln *P*:

$$\underset{(3.398)}{Seigp} > \underset{(3.053)}{Har} > \underset{(0.348)}{CIC1} > \underset{(0.327)}{RDF055p} > \underset{(0.320)}{HOMA} > \underset{(0.178)}{MATS1v} > \underset{(0.041)}{MAXDP} \qquad (6.1.4)$$

De la Ec. (6.1.2) se puede concluir que el aumento numérico de los descriptores *CIC1*, *RDF055p*, *HOMA* y *MAXDP* (con coeficientes positivos) y valores decrecientes de los descriptores *SEigp*, *Har* and *MATS1v* (con coeficientes negativos) tenderá a aumentar la fluorofilicidad.

El ordenamiento de contribuciones dado por la Ec. (6.1.4) indica que la distribución de la distancia topológica en las moléculas estudiadas, expresado por los descriptores tales como *SEigp* y *Har*, juegan un papel esencial que influye en los valores de ln *P*.

Cómo una aplicación práctica del modelo obtenido se predijeron las fluorofilicidades de compuestos que aún no han sido sintetizados en la actualidad, mostrados en la Tabla 6.1.3, los 69 compuestos están ordenados de acuerdo a los valores de fluorofilicidad. El análisis teórico muestra compuestos orgánicos con gran fluorofilicidad: $(R_{f8})_3P$ ($\ln P = 5.84$), 1,3,4-$(R_{f8})_3C_6H_3$ ($\ln P = 5.21$), *p*-$R_{f8}C_6F_4R_{f8}$ ($\ln P = 4.85$), que en principio pueden ser candidatos para ser sintetizados y usados en catálisis luego de ser verificados.

6.1.3 Métodos

Cómo es usual las moléculas fueron pre optimizadas con un método de *Molecular Mechanics Force Field* (MM+), y luego se refinó la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere y un límite de gradiente de 0.01 kcal.Å-1

Luego se ingresaron en el software Dragon[119] resultando un grupo de 1268 descriptores moleculares de distintas clases[120]. Adicionalmente diez descriptores constitucionales y cuatro derivados de la química-cuántica (momento dipolar molecular, energías totales y energías de HOMO-LUMO) no incluidos en el software Dragon se agregaron al conjunto total de descriptores. Los descriptores empíricos y basados en propiedades que provee el software fueron descartados.

Se usaron validaciones cruzadas (*l-n%-o*) [36], con n% representando el número de moléculas removidas del grupo de calibración, usando 100000 casos aleatorios.

6.1.4 Conclusiones

Se logró llevar a cabo un modelo que tiene un buen desempeño predictivo de la fluorofilicidad usando un grupo de calibración de 116 compuestos orgánicos.

Algunas de las moléculas incluidas en el análisis fueron sintetizadas recientemente[111, 115] y por lo tanto no usadas con anterioridad en un estudio similar.

Los parámetros estadísticos del presente modelo se comparan muy bien respecto a otros presentados anteriormente basados en LFER y MOD.[111, 113]

Los mejores descriptores teóricos que aparecen en la ecuación final pueden reflejar el tamaño molecular, simetría, aromaticidad, como así también la importancia del contenido de flúor en los compuestos estudiados. Adicionalmente se logró usar satisfactoriamente el modelo en moléculas que aún no han sido sintetizadas, dejando 3 candidatos para ser sintetizados y estudiados con mayor detalle.

La calidad de este trabajo tuvo como consecuencia su publicación en una reconocida revista del tema: A.G. Mercader, P.R. Duchowicz, M.A. Sanservino, F.M. Fernandez, and E.A. Castro, Journal of Fluorine Chemistry, 2007. 128(5): p. 484-492.

Tabla 6.1.1 Clasificación de los descriptores moleculares usados en los modelos QSPR.

Símbolo	Descripción	Tipo
MATS1v	Moran autocorrelation-lag 1/weighted by atomic van der Waals volumes	2D Autocorrelations
RDF055p	RDF-5.5 weighted by atomic polarizabilities	RDF [a]
MAXDP	maximal electrotopological positive variation	Topological
Har	Harary H index	Topological
CIC1	complementary information content (neighborhood symmetry of 1-order)	Topological
HOMA	armonic oscillator model of aromaticity index	Aromaticity Indices
SEigp	eigenvalue sum from polarizability weighted distance matrix	Topological
RTm	R total index/weighted by atomic masses	GETAWAY [b]
HATSp	Leverage-weighted total index/weighted by atomic polarizabilities	GETAWAY
PCD	Difference of multiple path counts to path counts	Topological
H2e	H autocorrelation of lag 2/weighted by atomic Sanderson electronegativities	GETAWAY
C-006	number of CH_2RX groups	Atom Centred-Fragments
N-068	number of Al_3-N groups [c]	Atom Centred-Fragments
GATS1p	Geary autocorrelation-lag 1/weighted by atomic polarizabilities	2D Autocorrelations

[a] RDF=Radial Distribution Function
[b] GETAWAY= GEometry, Topology and Atoms-Weighted AssemblY
[c] Al: grupos alifáticos

Tabla 6.1.2 Valores experimentales de fluorofilicidad, y predicciones realizadas con Ec. (6.1.2) y Huque et al. Los residuos se presentan en paréntesis

N°	Nombre del compuesto	Exp.	Ec. (6.1.2)	Huke et al
1	Decane	-2.86	-3.09(0.23)	-3.07(0.21)
2	Undecane	-3.13	-3.28(0.15)	-3.13(0.00)
3	Dodecane	-3.35	-3.47(0.12)	-3.19(-0.16)
4	Tridecane	-3.71	-3.67(-0.04)	-3.24(-0.47)
5	Tetradecane	-3.94	-3.87(-0.07)	-3.30(-0.64)
6	Hexadecane	-4.50	-4.30(-0.20)	-3.41(-1.09)
7	Dec-1-ene	-2.99	-3.50(0.51)	-3.29(0.30)
8	Undec-1-ene	-3.26	-3.65(0.39)	-3.34(0.08)
9	Dodec-1-ene	-3.66	-3.81(0.15)	-3.40(-0.26)
10	Tridec-1-ene	-3.94	-3.98(0.04)	-3.46(-0.48)
11	Tetradec-1-ene	-4.12	-4.17(0.05)	-3.51(-0.61)
12	Hexadec-1-ene	-4.70	-4.56(-0.14)	-3.62(-1.08)
13	$R_{f8}CH=CH_2$	2.67	1.65(1.02)	2.82(-0.15)
14	Cyclohexanone	-3.79	-3.87(0.08)	-3.96(0.17)
15	Cyclohexenone	-4.06	-4.57(0.51)	-4.25(0.19)
16	Cyclohexanol	-4.12	-4.31(0.19)	-4.74(0.62)
17	Trifluoroethanol	-1.77	-1.92(0.15)	-1.37(-0.40)
18	$(CF_3)_2CHOH$	-1.02	-1.07(0.05)	-0.70(-0.32)
19	$R_{f6}(CH_2)_2OH$	0.10	0.05(0.05)	0.47(-0.37)
20	$R_{f6}(CH_2)_3OH$	-0.24	-0.14(-0.10)	0.50(-0.74)
21	$R_{f8}(CH_2)_2OH$	1.02	1.47(-0.45)	0.72(0.30)
22	$R_{f8}(CH_2)_3OH$	0.59	1.16(-0.57)	0.80(-0.21)
23	$R_{f10}(CH_2)_3OH$	1.42	2.10(-0.68)	1.25(0.17)
24	Pentafluorobenzene	-1.24	-1.40(0.16)	-0.58(-0.66)
25	Hexafluorobenzene	-0.94	-0.34(-0.60)	-0.12(-0.82)
26	Ethylbenzene	-4.41	-3.31(-1.10)	-4.23(-0.18)
27	Dodecylbenzene	-4.70	-4.53(-0.17)	-4.79(0.09)
28	$R_{f8}(CH_2)_3C_6H_5$	-0.02	0.48(-0.50)	0.38(-0.40)
29	$o\text{-}R_{f6}(CH_2)_3C_6H_4(CH_2)_3R_{f6}$	1.03	1.20(-0.17)	1.37(-0.34)
30	$o\text{-}R_{f8}(CH_2)_3C_6H_4(CH_2)_3R_{f8}$	2.34	2.69(-0.35)	2.32(0.02)
31	$o\text{-}R_{f10}(CH_2)_3C_6H_4(CH_2)_3R_{f10}$	3.62	3.40(0.22)	3.23(0.39)
32	$m\text{-}R_{f8}(CH_2)_3C_6H_4(CH_2)_3R_{f8}$	2.28	2.96(-0.68)	2.32(-0.04)
33	$p\text{-}R_{f8}(CH_2)_3C_6H_4(CH_2)_3R_{f8}$	2.33	2.97(-0.64)	2.32(0.01)
34	$R_{f8}(CH_2)_3Cl$	0.03	0.74(-0.71)	0.37(-0.34)
35	$R_{f8}(CH_2)_3NH_2$	0.85	0.29(0.56)	1.29(-0.44)

36	$R_{f8}(CH_2)_3NH(CH_2)_3R_{f8}$	3.32	2.75(0.57)	3.34(-0.02)
37	$(R_{f6}(CH_2)_2)_3P$	4.41	4.46(-0.05)	3.75(0.66)
38	$(R_{f8}(CH_2)_3)_3P$	4.41	–	4.79(-0.38)
39	$(R_{f8}(CH_2)_4)_3P$	4.50	–	4.53(-0.03)
40	$(R_{f8}(CH_2)_5)_3P$	4.50	–	4.27(0.23)
41	$(R_{f6}(CH_2)_2)_2PC_{10}H_{19}$ (menthyl)	1.29	0.92(0.37)	1.11(0.18)
42	$(R_{f8}(CH_2)_2)_2PC_{10}H_{19}$ (menthyl)	2.70	1.92(0.78)	2.10(0.60)
43	$(p\text{-}R_{f6}C_6H_4)_3P$	-1.32	-1.31(-0.01)	-0.57(-0.75)
44	$(p\text{-}R_{f8}C_6H_4)_3P$	0.76	–	0.78(-0.02)
45	$Ph(CH_2)_2SiH_3$	-3.29	-3.22(-0.07)	-4.53(1.24)
46	$Ph(CH_2)_2SiOC_8H_{15}$	-5.11	-5.15(0.04)	-5.56(0.45)
47	$Ph(CH_2)_2SiOC_6H_{11}$ (cyclohexyl)	-4.82	-4.89(0.07)	-5.56(0.74)
48	$R_{f6}I$	1.31	1.53(-0.22)	0.34(0.97)
49	$R_{f8}I$	2.04	2.69(-0.65)	0.93(1.11)
50	$R_{f10}I$	2.84	2.43(0.41)	1.48(1.36)
51	$R_{f8}CH{=}CH_2$	2.67	1.92(0.75)	2.82(-0.15)
52	$R_{f8}(CH_2)_3SH$	0.24	1.04(-0.80)	1.23(-0.99)
53	$R_{f8}N(CH_2CH_2)_2$	0.86	0.91(-0.05)	1.48(-0.62)
54	$R_{f6}S(CH_2)_2CO_2Et$	-0.67	-0.17(-0.50)	-0.05(-0.62)
55	$R_{f8}S(CH_2)_2CO_2E$	0.04	0.76(-0.72)	0.49(-0.45)
56	CF_3SPh	-2.45	-2.87(0.42)	-2.01(-0.44)
57	$m\text{-}CF_3SC_6H_4CF_3$	-1.58	-2.17(0.59)	-0.85(-0.73)
58	$R_{f8}SPh$	0.59	1.03(-0.44)	-0.15(0.74)
59	$R_{f7}CH_2NHMe$	1.07	0.79(0.28)	1.49(-0.42)
60	$R_{f7}CH_2NMe_2$	1.53	1.10(0.43)	1.63(-0.10)
61	$R_{f7}CH_2N(CH_2CH_2)_2O$	0.14	0.43(-0.29)	0.60(-0.46)
62	$R_{f7}CH_2NHCH(Me)Ph$	-0.87	-0.73(-0.14)	-0.65(-0.22)
63	$R_{f7}C(O)Ph$	0.48	0.18(0.30)	–
64	$R_{f7}C(O)OCH_2Ph$	2.14	0.54(1.60)	–
65	$p\text{-}R_{f7}C(O)OCH_2C_6H_4OCF_3$	3.15	1.55(1.60)	–
66	$R_{f7}C(O)SMe$	1.16	0.92(0.24)	0.57(0.59)
67	$R_{f7}C(O)NHMe$	0.15	0.82(-0.67)	-0.23(0.38)
68	$R_{f7}C(O)NMe_2$	0.34	0.72(-0.38)	0.66(-0.32)
69	$R_{f7}C(O)N(CH_2CH_2)_2O$	-0.62	-0.32(-0.30)	-0.38(-0.24)
70	$R_{f7}C(S)Me$	1.08	1.46(-0.38)	0.19(0.89)
71	$R_{f7}C(S)NMe_2$	-0.66	0.22(-0.88)	-0.20(-0.46)
72	$R_{f7}C(S)N(CH_2CH_2)_2O$	-1.56	-1.06(-0.50)	-1.18(-0.38)
73	$R_{f7}C(S)NHCH(Me)Ph$	-1.84	-1.03(-0.81)	-3.18(1.34)
74	C_6H_6	-2.77	-2.58(-0.19)	-4.12(1.35)

75	CF_3Ph	-1.96	-2.39(0.43)	-1.82(-0.14)
76	$R_{f6}Ph$	0.54	0.33(0.21)	0.24(0.30)
77	$R_{f8}Ph$	1.24	1.38(-0.14)	0.78(0.46)
78	$R_{f10}Ph$	1.77	2.29(-0.52)	1.28(0.49)
79	o-$R_{f8}C_6H_4CF_3$	1.50	1.52(-0.02)	1.37(0.13)
80	m-$R_{f8}C_6H_4CF_3$	2.37	1.99(0.38)	1.37(1.00)
81	p-$R_{f8}C_6H_4CF_3$	2.13	2.01(0.12)	1.37(0.76)
82	p-$R_{f8}C_6H_4R_{f8}$	4.98	4.63(0.35)	–
83	[p-$CF_3C_6H_4(CF_2)_4]_2$	-0.56	-0.10(-0.46)	-0.18(-0.38)
84	o-$R_{f6}(CH_2)_2C_6H_4Cl$	-0.64	-0.99(0.35)	-0.63(-0.01)
85	p-$R_{f6}(CH_2)_2C_6H_4Cl$	-1.02	-1.02(0.00)	-0.63(-0.39)
86	p-$R_{f8}(CH_2)_2C_6H_4Cl$	-0.37	-0.05(-0.32)	-0.04(-0.33)
87	o-$R_{f6}(CH_2)_2C_6H_4Br$	-1.05	-1.12(0.07)	-1.22(0.17)
88	m-$R_{f6}(CH_2)_2C_6H_4Br$	-1.44	-1.09(-0.35)	-1.22(-0.22)
89	p-$R_{f6}(CH_2)_2C_6H_4Br$	-1.49	-1.13(-0.36)	-1.22(-0.27)
90	o-$R_{f8}C_6H_4CO_2Me$	-0.39	0.34(-0.73)	-0.18(-0.21)
91	m-$R_{f8}C_6H_4CO_2Me$	0.12	-0.11(0.23)	-0.18(0.30)
92	p-$R_{f8}C_6H_4CO_2Me$	-0.01	0.00(-0.01)	-0.18(0.17)
93	1,3,5-$R_{f8}C_6H_3(CF_3)_2$	4.05	2.70(1.35)	–
94	1,3,5-$(R_{f8})_2C_6H_3CO_2Me$	4.41	3.60(0.81)	–
95	1,3,5$(R_{f8})_2C_6H_3CH_2OH$	3.62	3.19(0.43)	–
96	1,3,5-$(R_{f8})_2C_6H_3CHO$	4.25	4.10(0.15)	–
97	2-$R_{f8}C_5H_4N$ (pyridine)	0.54	1.12(-0.58)	0.64(-0.10)
98	3-$R_{f8}C_5H_4N$ (pyridine)	0.88	1.02(-0.14)	0.64(0.24)
99	4-$R_{f8}C_5H_4N$ (pyridine)	0.80	1.28(-0.48)	0.64(0.16)
100	$(CF_3)_3CO(CH_2)_2NH_2$	-0.14	-0.46(0.32)	–
101	$(CF_3)_3CO(CH_2)_2NH(CH_3)$	-0.08	-0.19(0.10)	–
102	$[(CF_3)_3CO(CH_2)_2]_2NH$	1.82	2.02(-0.20)	–
103	$(CF_3)_3CO(CH_2)_2NH(CH_2)_3R_{f8}$	2.69	2.32(0.37)	–
104	$(CF_3)_3CO(CH_2)_2N(CH_3)_2$	0.34	0.25(0.08)	–
105	$[(CF_3)_3CO(CH_2)_2]_2NCH_3$	2.03	2.20(-0.17)	–
106	$[(CF_3)_3CO(CH_2)_2]_3N$	3.62	3.63(-0.01)	–
107	$R_{f8}(CH_2)_3NH_2$	0.85	0.29(0.56)	–
108	$R_{f8}(CH_2)_4NH_2$	0.54	0.40(0.14)	–
109	$R_{f8}(CH_2)_5NH_2$	0.28	0.24(0.04)	–
110	$R_{f7}CH_2NH(CH_3)$	1.07	0.79(0.28)	–
111	$R_{f8}(CH_2)_3NH(CH_3)$	0.88	0.76(0.12)	–
112	$[R_{f4}(CH_2)_3]_2NH$	0.71	0.91(-0.20)	–
113	$[R_{f6}(CH_2)_3]_2NH$	1.98	2.12(-0.14)	–

114	$[R_{f10}(CH_2)_3]_2NH$	4.08	4.28(-0.20)	–
115	$[R_{f8}(CH_2)_3]_2NH$	3.32	3.53(-0.21)	–
116	$[R_{f8}(CH_2)4]_2NH$	2.97	3.43(-0.46)	–
117	$[R_{f8}(CH_2)_5]_2NH$	2.59	2.88(-0.29)	–
118	$R_{f7}CH_2N(CH_3)_2$	1.53	1.10(0.43)	–
119	$R_{f8}(CH_2)_3N(CH_3)_2$	1.37	0.92(0.45)	–
120	$[R_{f8}(CH_2)_3]_2NCH_3$	3.63	3.28(0.35)	–

R_{fn} significa $(CF_2)_{n-1}CF_3$.

Tabla 6.1.3 Matriz de correlación para los descriptores de Ec. (6.1.2)

	SEigp	*RDF055p*	*MAXDP*	*Har*	*CIC1*	*MATS1v*	*HOMA*
SEigp	1	0.8059	0.7854	0.9738	0.4121	0.09542	0.09487
RDF055p		1	0.6357	0.8496	0.4684	0.2885	0.1510
MAXDP			1	0.7283	0.0242	0.1353	0.2478
Har				1	0.5284	0.1988	0.1868
CIC1					1	0.2766	0.09254
MATS1v						1	0.5124
HOMA							1

Tabla 6.1.4 Predicc ones de compuestos con fluorofilicidad desconocida

N°	*Nombre*	*Ec. (6.1.2)*
1	$(R_{f8})_3P$	5.84
2	$1,3,4\text{-}(R_{f8})_3C_6H_3$	5.21
3	$p\text{-}R_{f8}\,C_6F_4\,R_{f8}$	4.85
4	$R_{f11}CF{=}CF_2$	4.61
5	$R_{f9}CF_3$	4.47
6	$R_{f9}(CH_2)_2NH(CH_2)_2R_{f9}$	4.45
7	$p\text{-}R_{f8}\,C_6H_3F\,R_{f8}$	4.19
8	$1,3,5\text{-}(R_{f4})_2C_6H_3\,R_{f8}$	4.16
9	$(R_{f7}CH_2)_3P$	4.10
10	$m\text{-}R_{f16}C_6H_3CHO$	3.97
11	$m\text{-}R_{f16}C_6H_3CH_2OH$	3.69
12	$m\text{-}R_{f16}C_6H_3CO_2Me$	3.64
13	$1,3\text{-}(R_{f8})_2\text{-}5\text{-}CO_2CF_3C_6H_3$	3.05
14	$CH_3CH_2NH(CH_2)_3R_{f20}$	3.01
15	$1\text{-}R_{f16}\text{-}2,3\text{-}F_2\text{-}5\text{-}CH_2OH\,C_6H_2$	2.94
16	$m\text{-}R_{f17}C_6H_3CO_2Me$	2.55

17	$R_{f9}(CH_2)_2NH(CH_2)_2R_{f2}$	2.31
18	$(CF_3)_3CO(CF_2)_2NF_2$	2.18
19	*o*-$R_{f10}(CH_2)_3C_6H_4(CH_2)_3R_{f3}$	1.68
20	$CH_3(CH_2)_2NCH_3(CH_2)_3R_{f16}$	1.55
21	$R_{f8}(CH_2)_3NH(CH_2)_3CF_3$	1.07
22	$F_5C_6(CF_2)_2SiOC_8F_{15}$	0.95
23	$CF_3SC_6(CF_3)_5$	0.68
24	*o*-$R_{f3}(CH_2)_3C_6F_4(CH_2)_3R_{f2}$	0.55
25	*o*-$R_{f4}(CH_2)_3C_6H_4(CH_2)_3R_{f3}$	0.32
26	*o*-$R_{f3}(CH_2)_3C_6H_4(CH_2)_3R_{f3}$	-0.04
27	1,3-$(R_{f5})_2$-5-$(CH_2)_2SiOC_8H_{15}C_6H_3$	-0.12
28	*o*-$R_{f3}(CH_2)_3C_6H_4(CH_2)_3R_{f2}$	-0.47
29	1,3-$(R_{f5})_2$-2,4-F_2-5-$(CH_2)_2SiOC_8H_{15}C_6H$	-0.73
30	$R_{f8}C(S)NHCH(Me)Ph$	-0.76
31	Pentafluoroethanol	-0.81
32	7,10-R_{f4}-hexadec-1-ene	-1.63
33	1-R_{f7}-4-$(CH_2)_2SiOC_8H_{15}C_6H_4$	-2.06
34	$Ph(CH_2)_2SiF_3$	-2.08
35	4-F-1-$CF_3C_6H_4$	-2.35
36	1,3,5 Trifluorobenzene	-2.44
37	1,13-Difluorotridecane	-2.90
38	1-$(CF_2)_4CF_2H$-4-$(CH_2)_2SiOC_8H_{15}C_6H_4$	-2.92
39	6-fluoroundecane	-2.97
40	1,14-Difluorotetradecane	-3.09
41	1-Fluorododecane	-3.10
42	Cis-1,2-Difluorododec-1-ene	-3.13
43	6-fluorodec-1-ene	-3.24
44	2-fluoroundec-1-ene	-3.28
45	1,1-Difluorotridec-1-ene	-3.30
46	10-R_{f4}-hexadec-1-ene	-3.34
47	Propylbenzene	-3.41
48	$F_5C_6(CH_2)_2SiOC_8H_{15}$	-3.72
49	$F_5C_6(CF_2)_2SiOC_8H_{15}$	-3.82
50	$F_5C_6(CFH)_2SiOC_8H_{15}$	-3.88
51	Tetradec-1,13-ene	-4.33
52	PhSiOPh	-4.34
53	Hexadec-1,5,9,13-ene	-4.46
54	$CH_2{=}CH(CH_2)_{10}C_6H_5$	-4.60
55	1,2,3, 4-F_4-6-$(CH_2)_2SiOC_8H_{15}C_6H$	-4.67

56	$F_5C_6CFHCH_2SiOC_8H_{15}$	-4.67
57	$C_8H_{15}SiOC_8H_{15}$	-4.70
58	$Ph(CH_2)_2SiOPh$	-4.82
59	$1,2,3\text{-}F_3\text{-}5\text{-}(CH_2)_2SiOC_8H_{15}C_6H_2$	-4.83
60	2,5-Cyclohexadienone	-4.86
61	$1,2,3\text{-}F_3\text{-}4\text{-}(CH_2)_2SiOC_8H_{15}C_6H_2$	-4.88
62	3-Cyclohexenol	-5.01
63	$1,2\text{-}F_2\text{-}4\text{-}(CH_2)_2SiOC_8H_{15}C_6H_3$	-5.03
64	$1,2\text{-}F_2\text{-}3\text{-}(CH_2)_2SiOC_8H_{15}C_6H_3$	-5.04
65	Hexadec-1,3,5,7,9,11,13,15-ene	-5.11
66	*m*-F $C_6H_4(CH_2)_2SiOC_8H_{15}$	-5.18
67	Icosane	-5.21
68	*o*-F $C_6H_4(CH_2)_2SiOC_8H_{15}$	-5.24
69	1,3,5-Trihydroxybenzene	-5.43

R_{fn} significa $(CF_2)_{n-1}CF_3$

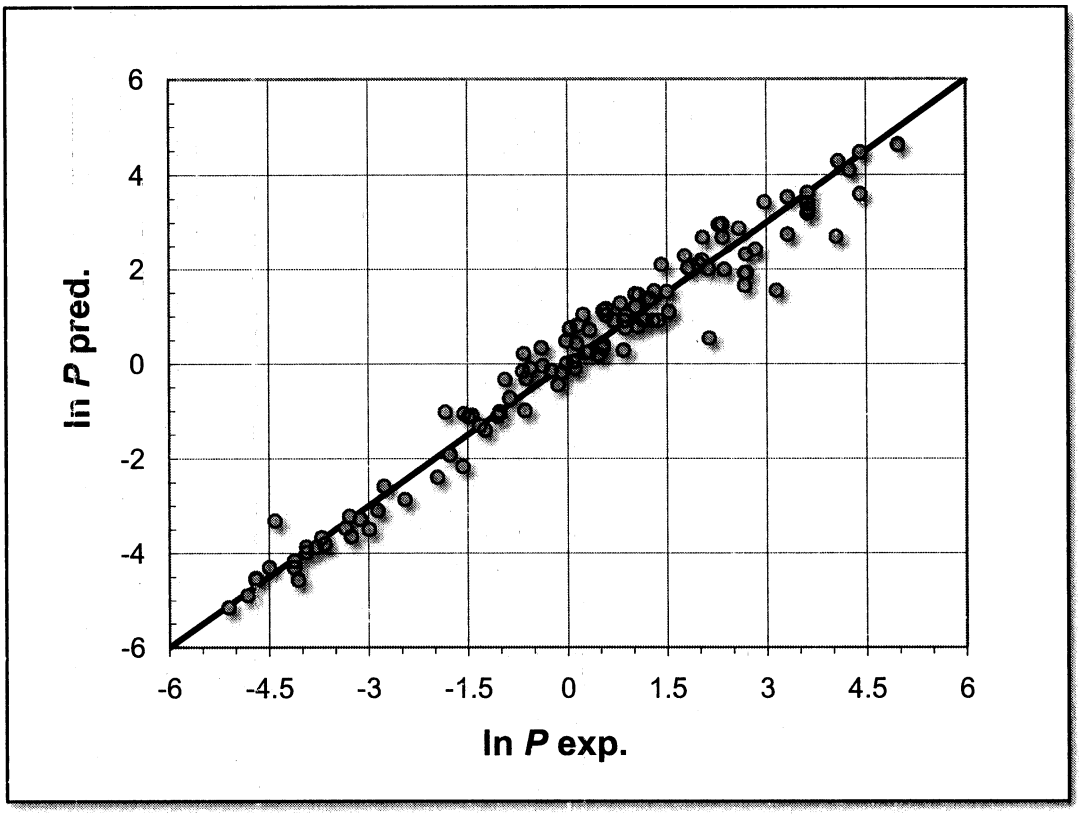

Figura 6.1.1 Fluorofilicidad predicha vs. experimental

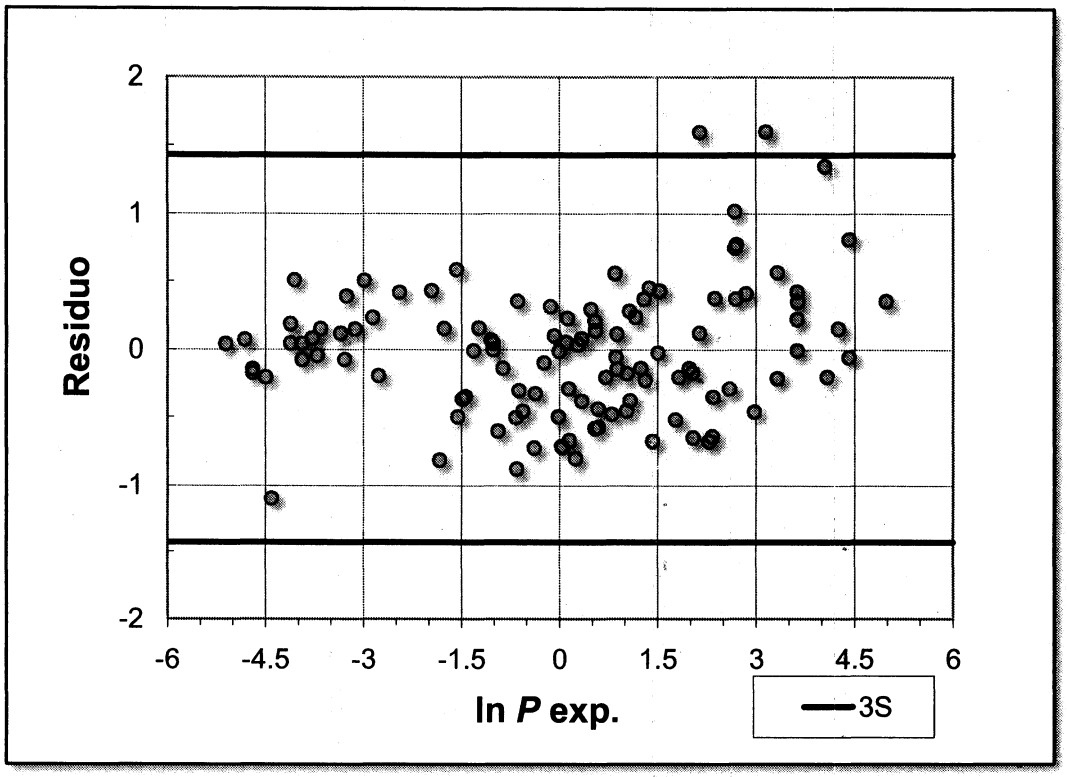

Figura 6.1.2 Gráfico de la dispersión de los residuos de la Ec. (6.1.2)

6.2 Predicción de la Toxicidad Acuosa de Derivados Heterogéneos de Fenol mediante QSAR

6.2.1 Introducción

Compuestos orgánicos con la estructura del fenol se han producido desde 1860 y están incluidos en un gran número de aplicaciones en diversas industrias como la textil, del cuero, papel, aceite y farmacéutica. Por ejemplo el ácido salicílico es usado en la producción de Aspirina y otros fármacos; los cloro-fenoles son usados para producir una gama de pesticidas; los fenoles alquílicos son usados en la producción de surfactantes y detergentes; el bis fenol es usado en la síntesis de resinas epoxi para pinturas, coberturas y molduras y en plásticos policarbonatos usados en discos compactos y electrodomésticos domésticos. Más allá de su gran importancia, la mayor desventaja de utilizar compuestos fenólicos es la contaminación de ecosistemas acuáticos y terrestres

resultante. Por lo tanto la estimación precisa de su impacto ambiental negativo posee un gran interés para la comunidad científica, en conjunto con una manera conveniente de controlar su producción.[121, 122]

Es sabido que llevar a cabo experimentos toxicológico para una sustancia determinada no es una tarea sencilla, usualmente resulta costosa tanto en tiempo como en dinero, además este tipo de análisis debe considerar múltiples entornos y todas las interacciones biológicas con los organismos vivos del ecosistema, datos que usualmente no se encuentran disponibles.[123] Una estrategia ampliamente aceptada para sobrellevar la falta de medidas experimentales en sistemas biológicos complejos es el análisis basado en QSAR,[3] el cual en el presente estudio será sobre la toxicidad acuática de derivados del fenol.

En la actualidad, la mayor cantidad de datos de toxicidad medidos en laboratorio por un método confiable y robusto son los de inhibición del crecimiento de *Tetrahymena pyriformis* [121]. Por lo tanto estos datos han sido sometidos a varios estudios QSAR[124-126] variando de modelos lineares simples basados en unos cuantos parámetros fisicoquímicos a herramientas más sofisticadas de optimización. Cronin et. al. en 2002 [124] usaron la técnica de *Partial Least Squares* (PLS) en 108 descriptores moleculares para analizar un grupo grande de 200 fenoles heterogéneos que mostraban simultáneamente diferentes modos de actividad toxicológica, como narcóticos polares (N=173), desacoplamiento respiratorio (N=19), pro-electrófilos (N=27), electrófilos débiles (N=27) y ciclos pro-redox (N=4). Los autores juzgaron la calidad de los modelos finales con un grupo externo de 50 compuestos. En 2004, Devillers et. al. [127] usó el mismo grupo de datos para desarrollar un modelo alternativo usando PLS que mejoró los resultados estadísticos del anterior, dando adicionalmente un modelo basado en un *Artificial Neural Network* (ANN) mostrando aún mejores resultados.

En este trabajo se propone un modelo alternativo QSAR de toxicidad acuosa para los mismos grupos de calibración y de validación de fenoles heterogéneos antes usados[127]. Se exploraran un número de descriptores moleculares mucho más grande, constituido por descriptores de todas las clases, de tipo rígido y flexible, haciendo uso del *Replacement Method* (RM)[77] para seleccionar el subconjunto de variables optimo.

6.2.2 Métodos

6.2.2.1 Conjunto de datos

La toxicidad acuosa se expresan como $pIGC_{50} = \log(IGC_{50}^{-1})$, siendo IGC_{50} la concentración en [mmol.l^{-1}] que produce una inhibición del crecimiento de 50% a la *Tetrahymena pyriformis* en régimen estático. El grupo de calibración está compuesto por las primeras 200 moléculas que se muestran en la Tabla 6.2.1, las restantes 50 moléculas forman el grupo de validación. Esta tabla además muestra los valores experimentales de $pIGC_{50}$ y los anteriormente reportados por Devillers et. al.[127]

6.2.2.2 Descriptores moleculares

Las moléculas fueron pre optimizadas con un método de *Molecular Mechanics Force Field* (MM+), y luego se refinó la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere y un límite de gradiente de 0.01 kcal.Å-1

Luego se ingresaron en el software Dragon[119] resultando un grupo de 1338 descriptores moleculares de distintas clases[120]. Adicionalmente 74 descriptores constitucionales que tiene en cuenta grupos funcionales y su posición en la molécula; y cuatro derivados de la química-cuántica (momento dipolar molecular, energías totales y energías de HOMO-LUMO) no incluidos en el software Dragon se agregaron al conjunto total de descriptores. Los descriptores empíricos y basados en propiedades que provee el software fueron descartados.

Se usaron validaciones cruzadas (*l-n%-o*) [36], con *n%=60%* (120 fenoles) siendo este el número de moléculas removidas del grupo de calibración, usando 5000000 casos aleatorios.

6.2.2.3 Búsqueda del mejor modelo

En los cálculos de uso el sistema de computación Matlab 5.0 [128] usando el método de búsqueda RM[77].

El RM se usó adicionalmente para crear descriptores flexibles lo que permitió mejorar las relaciones cuantitativas llegando a un modelo con calidad similar al presentado por Devillers et. al.[127] para el cual se usó un ANN, método que es matemáticamente más complicado haciéndolo difícil de interpretar.

Existen varias formas de construir descriptores flexibles, en este caso se usaron combinaciones lineales de descriptores rígidos

6.2.3 Resultados y Discusión

Inicialmente se aplicó el RM al grupo de calibración compuesto por los 200 derivados del fenol para encontrar el número óptimo de parámetros d_{opt} que participarían en la relación estructura-toxicidad. La Tabla 6.2.2 muestra los mejores modelos compuestos por 1 a 10 descriptores. En la misma se puede apreciar que el mejor modelo sería el denominado M8 sin embargo se optó por el M7 ya que se consideró que la mejora en los los parámetros esadísticos de validación era marginal y no justificaba la inlclusión de un descriptor adicional. Los detalles de las variables numéricas se encuentran en la Tabla 6.2.3.

El mejor modelo lineal encontrado es:

$$\begin{aligned} \mathrm{p}IGC_{50} &= -0.423(\pm 0.2) + 0.133(\pm 0.008)\cdot H\text{-}046 - 7.787(\pm 0.7)\cdot RBF \\ &\quad +0.588(\pm 0.2)\cdot O\text{-}060 + 0.988(\pm 0.2)\cdot nOH1,4 + 0.0428(\pm 0.005)\cdot S0K \\ &\quad +0.422(\pm 0.1)\cdot DISPp + 0.263(\pm 0.03)\cdot C\text{-}026 \\ N &= 200,\ R = 0.851,\ S = 0.442,\ F = 72,\ AIC = 0.212,\ FIT = 2.024,\ p<10^{-4} \\ R_{loo} &= 0.835,\ Sloo = 0.463,\ R_{l-60\%-o} = 0.730,\ S_{l-60\%-o} = 0.651 \\ N &= 50,\ Rval = 0.903,\ Svat = 0.392 \end{aligned} \qquad (6.2.1)$$

donde, los errores absolutos de los coeficientes de la regresión están en paréntesis, F es el cociente de Fisher, p la significancia del modelo, AIC el criterio de información de Akaike's, *loo* significa validación cruzada de *Leave-One-Out* y *val* indica los resultados del grupo de validación.

La Tabla 6.2.1 incluye los valores predichos de $\mathrm{p}IGC_{50}$ por la Ec. (6.2.1) para los 200 fenoles. El gráfico de valores de toxicidad predichos vs. experimentales mostrados en la Figura 6.2.1 indica que los 200 compuestos de calibración y 50 de validación siguen una línea recta. Los residuos en función de los datos predichos mostrados en la Figura 6.2.2 muestran que los mejores descriptores dados en la Ec.

(6.2.1) llevan a predicciones que tienden a seguir una distribución normal para la mayoría de los fenoles.

Esta figura incluye 3 *outliers* con residuos que exceden el valor de $2.5S$=1.105: compuestos **139** (2,4,6-Trinitrophenol, -1.37), **175** (Methoxyhydroquinone, 1.21), y **192** (4-Nitrophenol, 1.14); la presencia de estos *outliers* puede ser consecuencia de la selección de descriptores o algún motivo de naturaleza física, sin embargo esto no puede ser comprobado.

La matriz de correlación de la Tabla 6.2.4 muestra que los descriptores en la Ec. (6.2.1) no están altamente inter-correlacionados, y por lo tanto respaldan la presencia de todas las variables del modelo. El poder predictivo del modelo lineal es satisfactorio por la estabilidad en la inclusión y exclusión de compuestos, tal lo indican los parámetros S_{loo}=0.463 y $S_{l\text{-}60\%\text{-}o}$=0.651, y en especial por la raíz cuadrada del error medio (*rms*) del grupo de validación *rms*=0.418 y por la correlación R_{val}=0.903.

La estandarización de los coeficientes[74] de la regresión de la Ec. (6.2.1) permite asignar una importancia mayor a los descriptores que muestran un coeficiente estandarizado absoluto mayor (mostrado entre paréntesis):

$$\begin{matrix} H-046 > RBF > S0K > C\text{-}026 > nOH1,4 > DISPp > \text{O-}060 \\ (0.67)\quad (0.52)\quad (0.40)\quad (0.37)\quad (0.20)\quad (0.17)\quad (0.15) \end{matrix} \qquad (6.2.2)$$

Todos los descriptores poseen valores positivos para los 200 fenoles. Este ordenamiento revela que los parámetros estructurales que contribuyen en mayor grado a las toxicidades acuosas son *H-046*, *RBF*, *S0K*, y *C-026*. Por lo tanto considerando el signo de los coeficientes de la Ec. (6.2.1), las moléculas con valores crecientes de los descriptores *H-046*, *S0K*, y *C-026* y valores decrecientes de *RBF* tendrán mayores valores de $pIGC_{50}$ predichos, considerando que el resto de los descriptores tienen menor influencia en la toxicidad acuosa. Un mayor número de hidrógenos presentes en la estructura molecular, como se refleja en *H-046*, incrementaría el número de interacciones electroestáticas en sitios específicos del inhibidor. Esto es también cierto para el caso del número de fragmentos $R-CX-R$ codificados por *C-026*, a mayor número de átomos electronegativos mayor el número de interacciones polares. En resumen Ec. (6.2.1) muestra que los derivados del fenol con estructuras topológicas asimétricas, conteniendo un gran número de hidrógenos y hetero-átomos, con pocos enlaces con posibilidad de rotación, tenderán a ser más potentes en la inhibición de la *Tetrahymena pyriformis*.

La Tabla 6.2.5 resume las estadísticas del conjunto de calibración y el de validación por la referencia[127] donde se llevaron a cabo dos estudios, uno de tres componentes PLS y uno usando una red neuronal (ANN) de tres capas; dicha tabla incluye los parámetros estructurales que parecen en esos modelos. Hay un acuerdo considerable entre la Ec. (6.2.1) y el modelo lineal PLS, sin embargo esta última tiene un mejor desempeño en ambos grupos. Hay que notar sin embargo que el modelo propuesto en este trabajo tiene tan solo 7 descriptores en contraste con los 10 usados en el modelo PLS. Ambos modelos comparten aspectos estructurales relevantes: el número de sustituyentes OH en la posición 1,4 en el anillo aromático. Este indicador mostró una influencia importante en la descripción de la toxicidad acuática en el trabajo de Devillers et al. El presente trabajo confirma esta conclusión usando una metodología más simple basada en regresiones lineales por medio de RM.

La Tabla 6.2.5 muestra que el mejor modelo lineal QSAR dado por Ec. (6.2.1) es estadísticamente inferior al resultante de ANN, por lo que se decidió usar descriptores flexibles para mejorar la calidad de las predicciones, combinándolas con variables rígidas para llevar a cabo el procedimiento. El método es bastante simple, primero se definen las variables flexibles como los mejores modelos lineales de la Tabla 6.2.2, con d_{min}=2,...,d_{opt}=7 descriptores rígidos. Luego el primer ciclo de optimización consiste en agregar estos al conjunto total de descriptores **D** con D=1338, resultando **D'** con D'=1344 variables, y se busca el mejor modelo lineal con d=2, lo que da como resultado:

$$pIGC_{50} = -0.025(\pm 0.04) + 0.658(\pm 0.1)F1 + 0.376(\pm 0.1)F2$$
$$N = 200,\ R = 0.859,\ S = 0.426,\ F = 277,\ AIC = 0.187,\ FIT = 2.716,\ p < 10^{-4}$$
$$R_{loo} = 0.354,\ S_{loo} = 0.433,\ R_{l\text{-}60\%\text{-}o} = 0.790,\ S_{l\text{-}60\%\text{-}o} = 0.515 \qquad (6.2.3)$$

Aquí los descriptores flexibles *F1* y *F2* son *M6* y *M7*, respectivamente, y corresponden a los modelos lineales de seis y siete parámetros de la Tabla 6.2.2. En el segundo ciclo de optimización, el nuevo descriptores flexible *F3* definido por la Ec. (6.2.3) es agregado a **D'** resultando un nuevo **D"** con $D"$=1345 descriptores. Un tercer y un cuarto ciclo son posible y hay que decidir cuando parar las iteraciones.

Debe tenerse en cuenta qeu existe un serio riesgo de sobre-ajustar el modelo cuando se pasa de un ciclo al siguiente. Por lo tanto, el número total de ciclos debe seleccionarse cuidadosamente. Luego de llevar a cabo cuatro ciclos el mejor modelo encontrado es:

$$pIGC_{50} = -0.034(\pm 0.04) + 1.004(\pm 0.04) \cdot F5 + 0.328(\pm 0.1).nOH2$$
$$N = 200,\ R = 0.880,\ S = 0.394,\ F = 339,\ AIC = 0.160,\ FIT = 3.111,\ p < 10^{-4} \quad (6.2.4)$$
$$R_{loo} = 0.876,\ S_{loo} = 0.401,\ R_{l\text{-}60\%\text{-}o} = 0.812, S_{l\text{-}60\%\text{-}o} = 0.487$$

donde *F5* está definido en la Tabla 6.2.6. Los valores de $\mathbf{p}IGC_{50}$ predichos por la Ec. (6.2.4) para los 200 fenoles también se muestran en la Tabla 6.2.1. La Figura 6.2.3 muestra las toxicidades predichas vs. experimentales, mientras que la Figura 6.2.4 muestra los residuos en función de los datos predichos. Esta figura incluye cinco *outliers* de calibración que exceden el valor de 2.5*S*=0.985: compuestos **1** (-0.990), **136** (1.022), **139** (-1.286), **169** (1.010), **175** (1.481), and **198** (-1.173). Los parámetros estadísticos para el conjunto de validación *rms*=0.352 y R_{val}=0.880, son de calidad similar a los logrados con ANN en la Tabla 6.2.5.

El mejor modelo QSAR de la Ec. (6.2.4) Se basa en un total de 13 descriptores moleculares rígidos. Si se calcula el mejor modelo conteniendo *d*=13 variables seleccionadas de **D** por RM, se encuentra el modelo llamado M13 mostrado en la Tabla 6.2.5. Se puede apreciar que este modelo incluye demasiadas variables teniendo un mejor desempeño en el conjunto de calibración (*rms*=0.349) pero uno peor en el grupo de validación (*rms*=0.425), esto es básicamente la principal consecuencia de un sobre-ajuste. La metodología basada en descriptores flexibles permite el uso de muchos descriptores disminuyendo el *rms* en el conjunto de calibración, y simultáneamente en el de validación. El método presente basado en regresiones lineales es considerado como más simple que ANN.

Cómo una aplicación práctica del mejor modelo encontrado que se muestra en Ec. (6.2.4) se llevaron a cabo las predicciones de 74 estructuras sin medidas experimentales de $\mathbf{p}IGC_{50}$. Estas se enumeraron en la Tabla 6.2.7 y fueron propuestas modificando grupos funcionales de derivados fenólicos del grupo de moléculas usados en el trabajo.

6.2.4 Conclusiones

Se logró obtener un modelo QSAR lineal que codifica satisfactoriamente la información estructurar relacionada con la toxicidad acuosa de 250 derivados del fenol.

El uso de descriptores flexibles como combinación lineal de descriptores rígidos hizo posible mejorar los parámetros estadísticos de forma que estos son de una calidad comparable a los obtenidos por redes neuronales artificiales (ANN) y por PLS. La nueva metodología mejora los modelos lineales sin introducir sofisticaciones no lineales. A pesar de que los modelos de descriptores flexibles introducen un mayor número de variables rígidas siguen siendo más simples y mas fáciles de interpretar que los basados en métodos no lineales.

Los resultados obtenidos en este trabajo tuvieron como consecuencia su publicación en una importante revista: P. R. Duchowicz, A. G. Mercader*, F. M. Fernández, E. A. Castro, Chemometrics and Intelligent Laboratory Systems, 90 (2008) 97–107

Tabla 6.2.1 Valores experimentales y calculados de $pIGC_{50}$ de 250 compuestos derivados del fenol.

nº	Nombre	$pIGC_{50}$			
		exp.	Ec.(6.2.1)	Ec. (6.2.4)	ANN [127]
1	4-Hydroxyphenylacetic acid	-1.50	-0.42	-0.51	-1.06
2	3-Hydroxybenzyl alcohol	-1.04	-0.41	-0.60	-0.60
3	4-Carboxyphenol	-1.02	-0.40	-0.41	-0.59
4	-Hydroxy-4-methoxybenzyl alcohol	-0.99	-0.38	-0.46	-0.73
5	-Hydroxy-3-methoxybenzyl amine	-0.97	-0.26	-0.39	-0.59
6	4-Hydroxyphenethyl alcohol	-0.83	-0.52	-0.62	-0.45
7	3-Carboxyphenol	-0.81	-0.25	-0.32	-0.49
8	4-Hydroxybenzamide	-0.78	-0.33	-0.31	-0.40
9	-Hydroxy-3-methoxybenzyl alcohol	-0.70	-0.35	-0.47	-0.79
10	2,6-Dimethoxyphenol	-0.60	-0.21	-0.38	-0.50
11	,6-Tris(dimethylaminomethyl)phenol	-0.52	-0.36	-0.47	-0.04
12	Salicylic acid	-0.51	-0.16	-0.27	-0.54
13	2-Methoxyphenol	-0.51	-0.08	-0.13	-0.35
14	5-Methylresorcinol	-0.39	0.00	0.06	-0.25
15	4-Methylcyanophenol	-0.38	0.26	0.25	-0.09
16	3-Hydroxyacetophenone	-0.38	-0.22	-0.43	0.28
17	2-Ethoxyphenol	-0.36	-0.08	-0.20	-0.13
18	4-Acetylphenol	-0.30	-0.31	-0.48	0.30
19	3-Ethoxy-4-methoxyphenol	-0.30	0.02	-0.06	-0.21
20	2-Methylphenol	-0.29	0.20	0.11	-0.26

21	2-Hydroxybenzamide	-0.24	-0.10	-0.19	-0.02
22	Phenol	-0.21	-0.04	-0.04	-0.17
23	4-Methylphenol	-0.18	0.03	0.04	0.00
24	Hydroxy-3-methoxyphenethyl alcohol	-0.18	-0.28	-0.36	-0.66
25	3-Acetamidophenol	-0.16	0.06	0.07	-0.41
26	-Hydroxy-4-methoxybenzaldehyde	-0.14	0.05	0.02	0.11
27	-Hydroxy-3-methoxyacetophenone	-0.12	-0.09	-0.19	0.13
28	3,5-Dimethoxyphenol	-0.09	-0.33	-0.22	-0.27
29	2-Hydroxyethylsalicylate	-0.08	0.47	0.14	0.14
30	3-Methylphenol	-0.06	0.14	0.12	0.01
31	Methyl-3-hydroxybenzoate	-0.05	0.38	0.22	0.24
32	-Methoxy-4-hydroxybenzaldehyde	-0.03	0.03	-0.01	0.12
33	4-Hydroxy-3-methoxybenzonitrile	-0.03	0.10	0.03	0.16
34	3-Ethoxy-4-hydroxybenzaldehyde	0.01	0.06	-0.02	0.22
35	4-Fluorophenol	0.02	0.40	0.44	0.08
36	2-Cyanophenol	0.03	-0.02	-0.01	0.18
37	5-Fluoro-2-hydroxyacetophenone	0.04	0.34	0.22	0.70
38	2,4-Dimethylphenol	0.07	0.39	0.34	0.18
39	2-Hydroxyacetophenone	0.08	-0.14	-0.32	0.51
40	2,5-Dimethylphenol	0.08	0.40	0.38	0.27
41	Methyl-4-hydroxybenzoate	0.08	0.22	0.08	0.41
42	3,5-Dimethylphenol	0.11	0.26	0.27	0.36
43	4'-Hydroxypropiophenone	0.12	0.10	0.04	0.66
44	2,3-Dimethylphenol	0.12	0.44	0.38	0.16
45	3,4-Dimethylphenol	0.12	0.45	0.39	0.15
46	2-Ethylphenol	0.16	0.41	0.24	0.26
47	Syringaldehyde	0.17	-0.05	-0.05	-0.06
48	Salicylhydrazide	0.18	-0.13	-0.21	-0.12
49	2-Chlorophenol	0.18	0.64	0.64	0.26
50	4- Hydroxy-2-methylacetophenone	0.19	0.13	0.08	0.37
51	4-Ethylphenol	0.20	0.22	0.13	0.43
52	3-Ethylphenol	0.23	0.36	0.22	0.33
53	Salicylaldoxime	0.25	-0.16	0.22	0.31
54	2,3,6-Trimethylphenol	0.28	0.75	0.87	0.65
55	2,4,6-Trimethylphenol	0.28	0.63	0.48	0.66
56	2-Hydroxy-5-methylacetophenone	0.31	0.19	0.09	0.61
57	2-Bromophenol	0.33	0.75	0.82	0.75
58	5-Bromo-2-hydroxybenzyl alcohol	0.34	0.48	0.65	0.27
59	2,3,5-Trimethylphenol	0.36	0.77	0.74	0.75

60	3-Methoxysalicylaldehyde	0.38	0.03	0.04	0.10
61	Salicylhydroxamic acid	0.38	-0.17	-0.25	0.13
62	2-Chloro-5-methylphenol	0.39	0.87	0.85	0.80
63	4- Allyl-2-methoxyphenol	0.42	0.13	0.07	0.32
64	2-Hydroxybenzaldehyde	0.42	0.03	0.09	0.21
65	2,6-Difluorophenol	0.47	0.88	0.64	0.61
66	Ethyl-3-hydroxybenzoate	0.48	0.37	0.14	0.51
67	4-Cyanophenol	0.52	0.00	0.04	0.19
68	4-Propyloxyphenol	0.52	0.11	0.20	0.38
69	4-Chlorophenol	0.55	0.62	0.70	0.52
70	Ethyl-4-hydroxybenzoate	0.57	0.28	0.07	0.70
7	5-Methyl-2-nitrophenol	0.59	0.63	0.77	1.27
72	2- Bromo-4-methylphenol	0.60	0.90	0.90	1.06
73	2,4-Difluorophenol	0.60	1.00	0.97	0.41
74	3-Isopropylphenol	0.61	0.56	0.57	0.61
75	5-Bromovanillin	0.62	0.73	0.54	0.46
76	α,α,α-Trifluoro-4-cresol	0.62	0.21	0.35	0.67
77	Methyl-4-methoxysalicylate	0.62	0.44	0.24	0.34
78	4-Bromophenol	0.68	0.84	0.96	0.70
79	2-Chloro-4,5-dimethylphenol	0.69	1.12	1.08	0.79
80	4-Butoxyphenol	0.70	1.26	1.42	0.78
81	4-Chloro-2-methylphenol	0.70	0.88	0.89	0.90
82	3-tert-Butylphenol	0.73	0.72	0.72	0.93
83	2,6-Dichlorophenol	0.73	0.87	0.78	0.73
84	2-Methoxy-4-propenylphenol	0.75	1.00	0.82	0.96
85	3-Chloro-5-methoxyphenol	0.76	0.56	0.57	0.46
86	4-Chloro-3-methylphenol	0.80	0.83	0.85	0.64
87	2-Isopropylphenol	0.80	0.55	0.50	0.53
88	2,6-Dichloro-4- fluorophenol	0.80	1.38	1.32	0.87
89	4-Iodophenol	0.85	1.20	1.26	1.10
90	2,2'-Biphenol	0.88	0.66	0.71	0.23
91	4-tert-Butylphenol	0.91	0.60	0.64	0.97
92	3,4,5-Trimethylphenol	0.93	0.66	0.64	0.64
93	2',4,4' -Tetrahydroxybenzophenone	0.96	1.39	1.52	0.97
94	4-sec-Butylphenol	0.98	0.72	0.74	1.11
95	3- Hydroxydiphenylamine	1.01	1.30	1.42	0.81
96	4-Hydroxybenzophenone	1.02	0.91	1.01	0.86
97	2,4-Dichlorophenol	1.04	1.23	1.33	1.33
98	2,4,6-Tribromoresorcinol	1.06	1.28	1.15	1.23

99	Benzyl-4-hydroxyphenyl ketone	1.07	0.88	0.96	0.75
100	4-Chloro-3-ethylphenol	1.08	0.91	0.87	1.22
101	2-Phenylphenol	1.09	1.19	1.17	1.19
102	2,5-Dichlorophenol	1.13	1.04	1.19	1.41
103	3-Chloro-4-fluorophenol	1.13	1.23	1.11	0.67
104	3- Bromophenol	1.15	0.91	0.99	0.90
105	6-tert-Butyl-2,4-dimethylphenol	1.16	1.44	1.62	1.07
106	4-Chloro-3,5-dirnethylphenol	1.20	0.82	0.94	0.93
107	2-Hydroxybenzophenone	1.23	0.99	0.96	1.15
108	4-tert-Pentylphenol	1.23	1.03	1.04	1.43
109	4- Bromo-3,5-dimethylphenol	1.27	0.90	1.07	1.31
110	4- Bromo-6-chloro- 2-cresol	1.28	1.55	1.72	1.44
111	4-Cyclopentylphenol	1.29	2.06	1.34	1.08
112	2-tert-Butylphenol	1.29	0.72	1.07	0.81
113	2-tert-Butyl-4-methylphenol	1.30	1.10	1.47	1.13
114	2-Hydroxydiphenylmethane	1.31	1.04	1.15	1.22
115	Butyl-4-hydroxybenzoate	1.33	1.07	1.01	1.35
116	3-Phenylphenol	1.35	1.21	1.30	1.33
117	n-Pentyloxyphenol	1.36	0.96	1.15	1.25
118	2,4-Dibromophenol	1.40	1.32	1.51	1.74
119	2,4,6-Trichlorophenol	1.41	1.36	1.33	1.57
120	-Hydroxy-4-methoxybenzophenone	1.42	1.30	1.33	1.09
121	Isoamyl-4-hydroxybenzoate	1.48	1.39	1.39	1.60
122	3,5-Dichlorosalicylaldehyde	1.55	1.23	1.30	1.36
123	4-Cyclohexylphenol	1.56	2.52	1.62	1.54
124	3,5-Dichlorophenol	1.57	1.11	1.25	1.56
125	3,5-Di-tert-butylphenol	1.64	1.70	1.84	2.29
126	3,5-Dibromosalicylaldehyde	1.64	1.46	1.50	1.68
127	3,4-Dichlorophenol	1.75	1.44	1.37	1.30
128	4- Bromo-2,6-dichlorophenol	1.78	1.50	1.69	1.59
129	2,6- Di -tert-butyl-4-methylphenol	1.80	2.21	2.42	1.38
130	Chloro-2-isopropyl-5-methylphenol	1.85	1.50	1.55	1.81
131	2,4,6-Tribromophenol	2.03	1.39	1.47	2.08
132	4-Heptyloxyphenol	2.03	1.48	1.63	1.91
133	4-tert-Octylphenol	2.10	1.86	1.97	2.17
134	4-(4-Bromophenyl)phenol	2.31	2.05	2.13	1.99
135	3,5-Diiodosalicylaldehyde	2.34	1.65	1.75	2.22
136	2,3,5-Trichlorophenol	2.37	1.37	1.35	1.86
137	4-Nonylphenol	2.47	2.46	2.87	2.53

138	Nonyl-4-hydroxybenzoate	2.63	2.45	2.66	2.57
139	2,4,6-Trinitrophenol	-0.16	1.21	1.13	0.33
140	3,4-Dinitrophenol	0.27	0.89	1.00	0.64
141	2,6-Dinitrophenol	0.54	0.62	0.64	0.35
142	2,6- Dichloro-4-nitrophenol	0.63	1.44	1.37	0.62
143	2,5-Dinitrophenol	0.95	0.90	1.08	0.82
144	2,4-Dinitrophenol	1.08	0.89	1.05	0.48
145	2,6-Dinitro-4-cresol	1.23	0.95	0.94	1.29
146	4- Bromo-2-fluoro-6-nitrophenol	1.62	1.94	1.74	1.44
147	Pentafluorophenol	1.64	2.18	1.95	1.85
148	4,6-Dinitro-2-methylphenol	1.72	1.27	1.52	1.24
149	2,4-Dichloro-6-nitrophenol	1.75	1.74	1.63	1.41
150	Pentachlorophenol	2.05	2.45	2.40	1.94
151	2,3,5,6-Tetrachlorophenol	2.22	1.80	1.77	1.48
152	Pentabromophenol	2.66	2.54	2.57	2.59
153	2,3,4,5-Tetrachlorophenol	2.71	2.42	2.44	2.19
154	4-Acetamidophenol	-0.82	-0.08	-0.15	-0.57
155	3-Aminophenol	-0.52	0.07	-0.04	-0.48
156	4-Aminophenol	-0.08	-0.06	-0.14	0.03
157	3-Methylcatechol	0.28	0.20	0.27	-0.18
158	2-Amino-4-tert-butylphenol	0.37	0.90	0.97	0.47
159	4-Methylcatechol	0.37	0.16	0.13	-0.21
160	1,2,4-Trihydroxybenzene	0.44	0.97	0.69	0.81
161	Hydroquinone	0.47	0.63	0.48	1.03
162	Catechol	0.75	-0.13	0.19	-0.40
163	2-Amino-4-chlorophenol	0.78	0.82	0.84	0.27
164	1 ,2,3-Trihydroxybenzene	0.85	-0.11	0.01	-0.52
165	2-Aminophenol	0.94	0.10	0.10	0.09
166	4-Chlorocatecol	1.06	0.82	1.16	0.45
167	Chlorohydroquinone	1.26	1.66	1.35	1.61
163	4-Amino-2-cresol	1.31	0.23	0.58	1.39
169	2,3-Dimethylhydroquinone	1.41	0.38	0.40	1.49
170	4- Amino-2,3-dimethylphenol	1.44	0.61	0.59	0.97
171	Bromohydroquinone	1.68	1.71	1.52	1.90
172	Tetrachlorocatechol	1.70	1.95	1.95	1.54
173	Phenylhydroquinone	2.00	1.35	1.76	1.73
174	3,5 -Di-tert-butylcatechol	2.11	2.17	2.00	1.87
175	Methoxyhydroquinone	2.20	0.99	0.72	0.94
176	3- Hydroxy-4-nitrobenzaldehyde	0.27	0.47	0.58	0.89

177	5- Hydroxy-2-nitrobenzaldehyde	0.33	0.41	0.45	0.69
178	2-Amino-4-nitrophenol	0.47	0.49	0.41	0.60
179	4- Methyl-2-nitrophenol	0.57	0.49	0.41	1.39
180	4- Hydroxy-3-nitrobenzaldehyde	0.61	0.47	0.56	0.67
181	4- Nitrosophenol	0.65	0.11	0.26	0.13
182	2-Nitroresorcinol	0.66	0.32	0.45	0.24
183	4-Methyl-3-nitrophenol	0.74	0.62	0.74	1.22
184	2-Chloromethyl-4-nitrophenol	0.75	0.65	0.74	0.84
185	romo-2'-hydroxy-5'-nitroacetanilide	0.87	1.18	1.05	1.31
186	4-Amino-2-nitrophenol	0.88	0.49	0.47	1.35
187	2-Fluoro-4-nitrophenol	1.07	0.88	0.92	0.71
188	5-Fluoro-2-nitrophenol	1.13	1.00	1.07	1.23
189	4-Nitrocatechol	1.17	0.41	0.84	0.78
190	2-Amino-4-chloro-5-nitrophenol	1.17	1.22	1.31	1.33
191	4- Fluoro-2-nitrophenol	1.38	1.01	1.07	1.25
192	4-Nitrophenol	1.42	0.28	0.47	0.89
193	2-Chloro-4-nitrophenol	1.59	1.14	1.19	0.92
194	4-Chloro-6-nitro-3-cresol	1.64	1.34	1.45	1.52
195	3- Methyl-4-nitrophenol	1.73	0.67	0.76	1.23
196	4-Bromo-2-nitrophenol	1.87	1.37	1.47	1.51
197	4-Chloro-2-nitrophenol	2.05	1.24	1.32	1.66
198	Tetrabromocatechol	0.98	2.08	2.15	1.75
199	Tetramethylhydroquinone	1.28	0.87	0.90	1.13
200	Tetrachlorohydroquinone	2.11	2.20	1.82	1.95
201	1,3,5-Trihydroxybenzene	-1.26	-0.41	-0.28	-0.66
202	2- Hydroxybenzylalcohol	-0.95	-0.28	-0.64	-0.64
203	Resorcinol	-0.65	-0.12	-0.07	-0.43
204	4-(4- Hydroxyphenyl)- 2-butanone	-0.50	-0.03	0.05	-0.12
205	3-Methoxyphenol	-0.33	-0.11	-0.10	-0.30
206	yl-4-hydroxy-3-methoxyphenylacetate	-0.23	0.53	0.39	-0.19
207	4-Methoxyphenol	-0.14	-0.27	-0.23	-0.28
208	3-Cyanophenol	-0.06	0.03	0.07	0.26
209	4-Ethoxyphenol	0.01	-0.24	-0.21	-0.11
210	4- Hydroxypropiophenone	0.05	0.15	0.11	0.59
211	3-Hydroxybenzaldehyde	0.09	0.02	0.07	0.20
212	4-Chlororesorcinol	0.13	0.77	0.73	0.22
213	2-Fluorophenol	0.19	0.56	0.48	0.09
214	4-Hydroxybenzaldehyde	0.27	-0.11	-0.04	0.13
215	2-Allylphenol	0.33	0.22	0.18	0.23

216	3-Fluorophenol	0.38	0.53	0.52	0.13
217	4-Isopropylphenol	0.47	0.46	0.46	0.70
218	-Hydroxy-4-methoxyacetophenone	0.55	-0.11	-0.17	0.38
219	3-Methyl-2-nitrophenol	0.61	0.64	0.72	1.18
220	4-Propylphenol	0.64	0.50	0.54	0.90
221	Hydroxy-4,5-dimethylacetophenone	0.71	0.50	0.45	0.81
222	2- Methyl-3-nitrophenol	0.78	0.69	1.14	1.52
223	3-Chlorophenol	0.87	0.70	0.74	0.62
224	4,6-Dichlororesorcinol	0.97	0.96	1.08	0.86
225	4-Benzyloxyphenol	1.04	0.93	1.12	0.58
226	3- Iodophenol	1.12	1.19	1.25	1.10
227	4- Bromo-2,6-dimethylphenol	1.17	1.06	1.27	0.93
228	2,3-Dichlorophenol	1.28	1.34	1.29	0.98
229	5- Pentylresorcinol	1.31	1.26	1.46	1.17
230	4-Phenylphenol	1.39	1.13	1.24	1.27
231	Benzyl-4-hydroxybenzoate	1.55	1.43	1.40	1.16
232	4-Hexyloxyphenol	1.64	1.12	1.39	1.61
233	4-Hexylresorcinol	1.80	1.70	1.67	1.60
234	2,4,5-Trichlorophenol	2.10	1.78	1.82	1.82
235	2-Ethylhexyl-4'-hydroxybenzoate	2.51	2.56	2.56	2.45
236	2,3-Dinitrophenol	0.46	0.96	1.06	0.56
237	2,3,5,6-Tetrafluorophenol	1.17	1.74	1.47	0.96
238	2,6-Diiodo-4-nitrophenol	1.71	1.82	1.81	1.58
239	3,4,5,6- Tetrabromo-2-cresol	2.57	2.69	2.88	2.54
240	2,4-Diaminophenol	0.13	-0.02	-0.19	-0.25
241	5- Amino-2-methoxyphenol	0.45	0.00	-0.13	-0.73
242	6- Amino-2,4-dimethylphenol	0.89	0.58	0.38	0.28
243	Trimethylhydroquinone	1.34	1.86	1.71	0.95
244	Methylhydroquinone	1.86	1.20	1.24	1.26
245	3-Nitrophenol	0.51	0.43	0.60	1.06
246	2-Nitrophenol	0.67	0.53	0.62	0.91
247	3-Fluoro-4-nitrophenol	0.94	0.92	1.03	0.97
248	2,6-Dibromo-4-nitrophenol	1.36	1.57	1.55	0.93
249	4-Nitro-3-(trifluoromethyl)phenol	1.65	0.95	0.95	1.77
250	Tetrafluorohydroquinone	1.84	2.20	1.59	2.26

Tabla 6.2.2. Modelos lineales QSAR encontrados para los 200 fenoles del conjunto de calibración. El mejor modelo encontrado se remarcó en negrita.

Modelo	descriptores usados	*R*	*S*	*FIT*	S_{loo}	$S_{l\text{-}60\%\text{-}o}$
M1	*Mor03p*	0.621	0.649	0.620	0.656	0.807
M2	*RDF020u, Mor03p*	0.695	0.597	0.903	0.608	0.770
M3	*MATS2m, RBF, RDF025u*	0.748	0.553	1.191	0.566	0.781
M4	*ATS2p, RDF020u, C-026, MATS8p*	0.770	0.530	1.298	0.553	0.763
M5	*MATS1m, C-024, C-026, nCOOHPh, H-046*	0.820	0.479	1.769	0.494	0.664
M6	*H-046, RBF, C-026, R3e+, TI2, MATS1m*	0.833	0.464	1.857	0.482	0.663
M7	***H-046, RBF, O-060, nOH1,4, S0K, DISPp, C-026***	**0.851**	**0.442**	**2.024**	**0.463**	**0.651**
M8	*H-046, RBF, C-026, TI2, R3e$^+$, R2v$^+$, JGI2, n2,6-P*	0.862	0.428	2.087	0.448	0.635
M9	*MATS1e, nCNPh, C-026, Mor02v, R3e$^+$, R2v$^+$, G(O..O), nRORPh, nOH1,4*	0.872	0.414	2.144	0.440	0.710
M10	*T(O..O), nRORPh, C-026, RDF025u, R3e$^+$, nCNPh, n2,6-P, HATS4u, R2v$^+$, MATS1e*	0.881	0.401	2.184	0.430	0.739

Tabla 6.2.3 Símbolos de los descriptores moleculares presentes en los distintos modelos.

Descriptor molecular	Tipo	Descripción
Mor03p	3D-MoRSE	3D-MoRSE – signal 03 / weighted by atomic polarizabilities
RDF020u	Radial Distribution Function	Radial distribution function – 2.0 / unweighted
MATS2m	2D-Autocorrelations	Moran autocorrelation lag-2 / weighted by atomic masses
RBF	Constitutional	Rotatable bond fraction
RDF025u	Radial Distribution Function	Radial distribution function – 2.5 / unweighted
ATS2p	2D-Autocorrelations	Broto-Moreau autocorrelation of a topological structure – lag 2 / weighted by atomic polarizabilities
C-026	Atom-Centred Fragments	number of R—CX—R[1]
MATS8p	2D-Autocorrelations	Moran autocorrelation lag-8 / weighted by atomic polarizabilities
MATS1m	2D-Autocorrelations	Moran autocorrelation lag-1 / weighted by atomic masses
C-024	Atom-Centred Fragments	number of R—CH—R
nCOOHPh	Functional Groups	number of carboxylic acids (aromatic)
H-046	Atom-Centred Fragments	H attached to C0(sp^3) with no X attached to next C[2]
R3e$^+$	GETAWAY	R maximal autocorrelation of lag 3 / weighted by atomic

		Sanderson electronegativities
TI2	Topological	second Mohar index
O-060	Atom-Centred Fragments	number of Al-O-Ar / Ar-O-Ar / R..O..R / R-O-C=X fragments[3]
nOH1,4	Constitutional	number of OH substituting position 1,4 in Ar
DISPp	Geometrical	d COMMA2 value / weighted by atomic polarizabilities
DISPv	Geometrical	d COMMA2 value / weighted by atomic volumes
S0K	Topological	Kier symmetry index
$R2v^+$	GETAWAY	R maximal autocorrelation of lag 2 / weighted by atomic van der Waals volumes
JGI2	Galvez Topological Charge Indices	Mean topological charge index of order 2 /
n2,6-P	Constitutional	Number of 2,6-substituted phenols
MATS1e	2D-Autocorrelations	Moran autocorrelation lag-1 / weighted by atomic Sanderson electronegativities
nCNPh	Functional Groups	Number of nitriles (aromatic)
Mor02v	3D-MoRSE	3D-MoRSE – signal 02 / weighted by atomic van der Waals volumes
G(O..O)	Geometrical	Sum of geometrical distances between O..O
nRORPh	Functional Groups	Number of ethers (aromatic)
T(O..O)	Topological	Sum of topological distances between O..O
HATS4u	GETAWAY	Leverage-weighted autocorrelation of lag 4 / unweighted
MW	Constitutional	Molecular Weight
logD	Empirical	Distribution Coefficient
ABSQon	Charge	Sum of absolute charges on O and N
SsOH	Electrotopological	Electrotopological state index for OH group
LUMO	Quantum Chemical	Energy of the lowest unoccupied molecular orbital
Pneg	Charge	Negatively charged molecular surface area
MaxHp	Charge	Largest positive charge on H atom
$4\text{-}NH_2$	Constitutional	Number of amino groups in para position
$nCrH_2$	Functional Groups	Number of ring secondary $C(sp^3)$
Mor22m	3D-MoRSE	3D-MoRSE – signal 22 / weighted by atomic masses
nOH2	Constitutional	Number of OH substituting position 2 in Ar
nCH	Constitutional	Number of CH fragments
Mor24e	3D-MoRSE	3D-MoRSE – signal 24 / weighted by atomic Sanderson electronegativities
TE1	Charge	Topographic electronic descriptor
GATS2e	2D-Autocorrelations	Geary autocorrelation of lag 2 / weighted by atomic Sanderson electronegativities

PW3	Topological	Path/walk 3-Randić shape index
C-040	Atom-Centred Fragments	number of R-C(=X)-X, R-C#X, X-=C=X
R_{ww}	Topological	Reciprocal hyper-detour index
F1	Flexible	*M6* from Table 2
F2	Flexible	*M7* from Table 2

[1]R representa cualquier grupo unido por un carbón. X=O, N, S, P, Se o alógeno.

[2]C0(sp^3)= átomo de carbono con número de oxidación 0 e hibridación sp^3.

[3]Al=Alifático, Ar=aromático.

Tabla 6.2.4. Matriz de correlación para los descriptores de la Ec. (6.2.1) (N=200).

Símbolo	***H-046***	***RBF***	***O-060***	***nOH1,4***	***S0K***	***DISPp***	***C-026***
H-046	1	0.342	0.022	0.099	0.094	0.111	0.379
RBF		1	0.313	0.002	0.236	0.202	0.301
O-060			1	0.038	0.201	0.081	0.196
nOH1,4				1	0.137	0.065	0.173
S0K					1	0.095	0.256
DISPp						1	0.289
C-026							1

Tabla 6.2.5 Parámetros estadísticos para los diferentes modelos QSAR de $pIGC_{50}$.

Modelo	**Descriptores**	***R***	***rms* calibración[1]**	R_{val}	***rms* validación**
PLS	*nOH1,4, logD, MW, ABSQon, SsOH, n2,6-P, LUMO, Pneg, MaxHp, 4-NH_2*	0.869	0.409	0.871	0.411
ANN	*nOH1,4, logD, MW, ABSQon, n2,6-P, LUMO, Pneg, MaxHp, 4-NH_2*	0.905	0.351	0.908	0.352
Ec. (6.2.1)	*H-046, RBF, O-060, nOH1,4, S0K, DISPp, C-026*	0.851	0.433	0.903	0.418
Ec. (6.2.4)	*F5, nOH2*	0.880	0.392	0.880	0.352
M13	*G(O..O), GATS2e, C-026, R3e$^+$, O-060, PW3, C-040, n2,6-P, R_{ww}, Mor24e, nOH1,4, DISPv, RBF*	0.906	0.349	0.863	0.425

[1] *rms*: raíz cuadrada del residuo medio

Tabla 6.2.6 Definiciones de los descriptores flexibles usados en el presente análisis.

Ecuación lineal
$F1 = -7.925 + 0.123 \cdot H\text{-}046 - 6.692.RBF + 0.307.C\text{-}026 - 7.617.R3e^{+} + 0.407.TI2 + 9.316.MATS1m$
$F4 = -0.011 + 1.027 \cdot F3 - 0.197.nCrH_2$
$F5 = 0.00227 + 1.056 \cdot F4 - 0.223 \cdot n2,6\text{-}P$

Tabla 6.2.7. Valores predichos de $pIGC_{50}$ para 74 derivados del fenol aún no medidos.

nº	Nombre	Ec. (6.2.4)	nº	Nombre	Ec. (6.2.4)
1	3,5-Diiodophenol	1.79	38	2-Fluoro-tetrachlorophenol	2.47
2	3,4,5-Triiodophenol	2.47	39	2-Bromo-tetrachlorophenol	2.49
3	2,3,4,5-Tetraiodophenol	2.90	40	2-Iodo-tetrachlorophenol	2.55
4	Pentaiodophenol	2.91	41	3-Bromo-tetrachlorophenol	2.53
5	1,3,5-Triiodophenol	1.81	42	4-Bromo-tetrachlorophenol	2.45
6	Tetraiodohydroquinone	2.12	43	3,4-Dibromo-trichlorophenol	2.66
7	Triiodo-1,2,4-benzenetriol	1.74	44	3,4,5-Tribromo-dichlorophenol	2.58
8	Diiodo-1,2,4,5-benzenetetraol	0.46	45	2,3,4,6-Tetrabromo-chlorophenol	2.68
9	Iodo-1,2,3,4,5-benzenepentaol	0.23	46	2,3,5,6-Tetrabromo-chlorophenol	2.52
10	3,4-Dihydroxyphenylacetic acid	-0.34	47	2,3,5,6-Tetrabromo-fluorophenol	2.32
11	3,4,5-Trihydroxyphenylacetic acid	-0.52	48	2,3,5,6-Tetrabromo-iodophenol	2.71
12	2,3,4,5-Tetrahydroxyphenylacetic acid	-0.01	49	Nonyl-3,4-dihydroxybenzoate	2.89
13	2,3,4,5,6-Pentahydroxyphenylacetic acid	-0.37	50	Nonyl-2,4,5-trihydroxybenzoate	3.91
14	4-Hydroxybenzene-1,3-biacetic acid	-0.43	51	Nonyl-2,3,4,6-tetrahydroxybenzoate	3.81
15	2-Hydroxybenzene-1,3,5-briacetic acid	-0.49	52	Nonyl-2,3,4,5,6-pentahydroxybenzoate	3.39
16	3-Hydroxyphenylacetic acid	-0.44	53	2,4-$CH_3C_8H_2OCO$-phenol	5.95
17	2-Hydroxyphenylacetic acid	-0.15	54	2,4,6-$CH_3C_8H_2OCO$-phenol	7.60
18	1,2,3,5-Tetrahydroxybenzene	0.45	55	2,5-Dimethyl-3,4,6-tribromophenol	2.37

19	Pentahydroxybenzene	0.20	56	2,4,5-Trimethyl-3,6-dibromophenol	2.11
20	Hexahydroxybenzene	-0.69	57	2,3,4,5-Tetramethyl-6-dibromophenol	1.84
21	3,4-Dihydroxybenzyl alcohol	-0.54	58	Pentamethylphenol	1.47
22	3,4,5-Trihydroxybenzyl alcohol	-0.72	59	2-Methyl-3,4,5,6-tetrachlorophenol	2.44
23	2,3,4,5-Tetrahydroxybenzyl alcohol	-0.29	60	2-Methyl-3,5-dibromo-4,6-dichlorophenol	2.62
24	Pentahydroxybenzyl alcohol	-0.59	61	2-Methyl-3,4,5-tribromo-6-chlorophenol	2.70
25	2-Chloro-3,4,5,6-tetrahydroxybenzyl alcohol	0.10	62	2-Methyl-3,4-dibromo-5,6-dichlorophenol	2.64
26	2,4-Dichloro-3,5,6-trihydroxybenzyl alcohol	0.80	63	2-Methyl-3,4-dibromo-5,6-diiodophenol	2.99
27	2,4,6-Trichloro-3,5-dihydroxybenzyl alcohol	0.60	64	2-Methyl-3-bromo-4,5,6-triiodophenol	3.07
28	2,3,4,6-Tetrachloro-5-hydroxybenzyl alcohol	1.40	65	2-Methyl-3-bromo-4,5-diiodo-6-chlorophenol	2.93
29	2,3,5,6-Tetrachloro-4-hydroxybenzyl alcohol	1.26	66	2-Methyl-3-bromo-4-iodo-5-fluoro-6-chlorophenol	2.65
30	4-Hydroxybenzene-1,3-dimethanol	-0.83	67	2-Methyl-3,4-dibromo-5-fluoro-6-chlorophenol	2.47
31	4-Carboxycatechol	-0.22	68	2-Methyl-3-bromo-4,5,6-trifluorophenol	2.13
32	1,2,3-Trihydroxy-5-carboxybenzene	-0.42	69	2,6-Dimethyl-3-bromo-4,5-difluorophenol	1.93
33	1,2,3,4-Tetrahydroxy-5-carboxybenzene	0.07	70	2,4,6-Trimethyl-3-bromo-5-fluorophenol	2.16
34	Pentahydroxy-carboxybenzene	-0.28	71	2,4,6-Trimethyl-3-bromo-5-chlorophenol	2.26
35	2,4-Dicarboxyphenol	-0.07	72	2,4,6-Trimethyl-3-fluoro-5-chlorophenol	2.05
36	2,4,6-Tricarboxyphenol	-0.67	73	2,4,6-Trimethyl-3-iodo-5-chlorophenol	2.35
37	1,3,5-Tetrahydroxy-2,4,6-tricarboxybenzene	-0.99	74	2,4,6-Trimethyl-3,5-diiodophenol	2.45

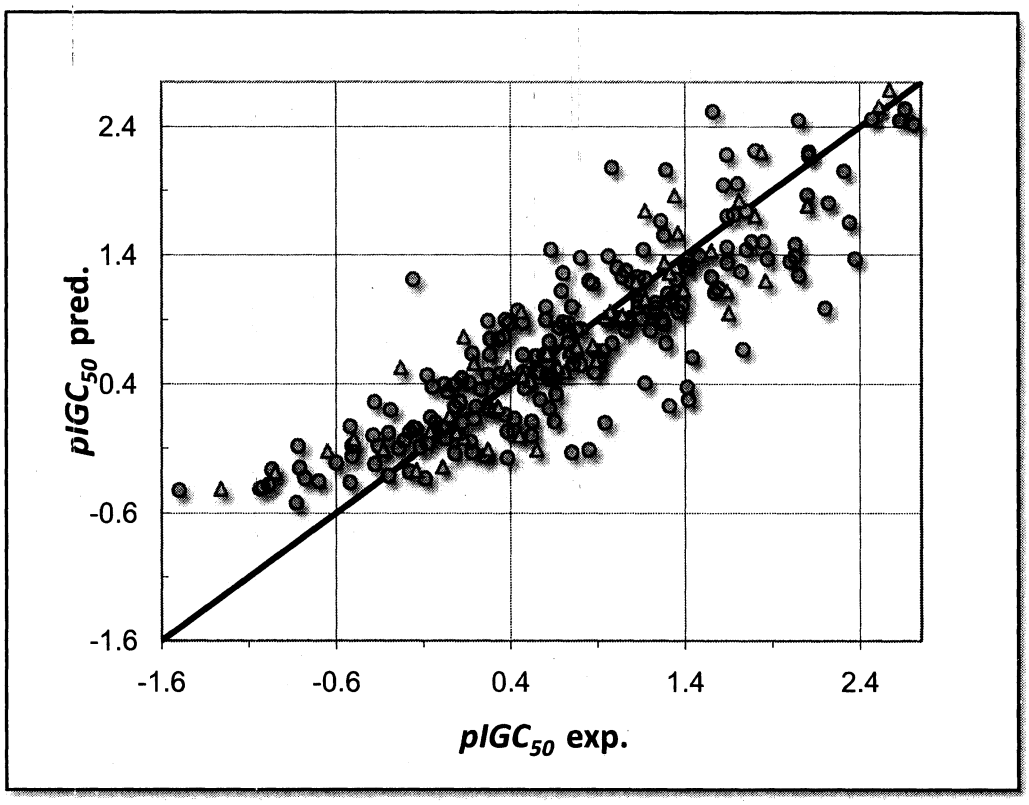

Figura 6.2.1 Valores predichos por Ec.(6.2.1) versus experimentales de pIGC_{50}

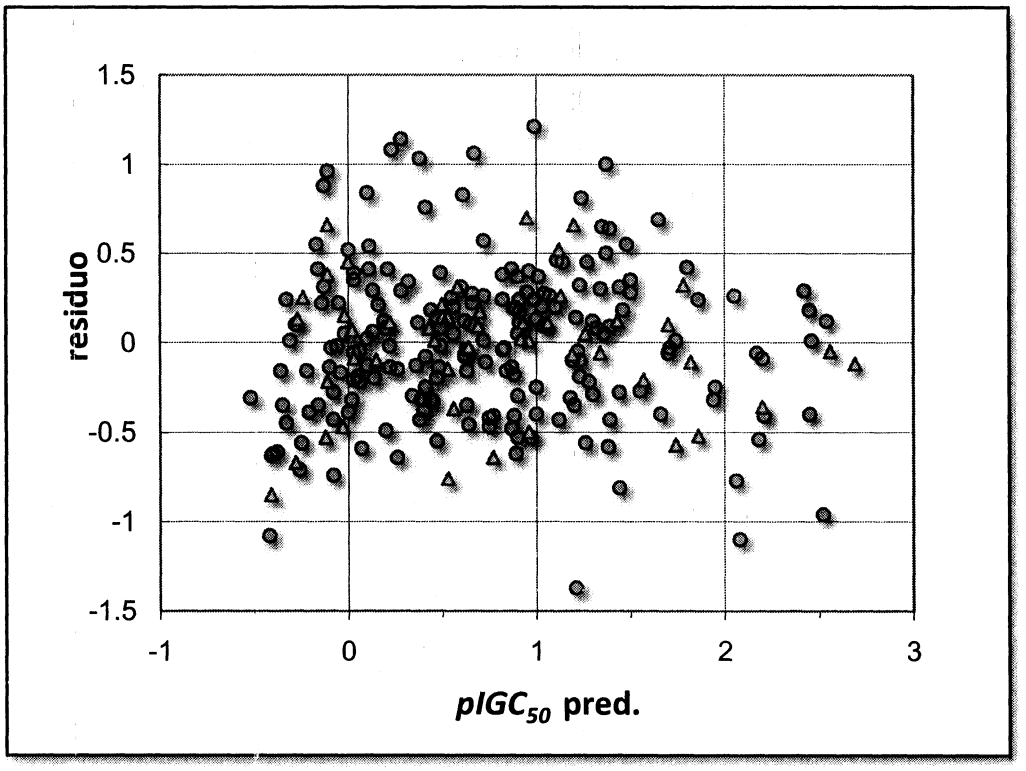

Figura 6.2.2 Gráfico de la dispersión de residuos para la Ec.(6.2.1).

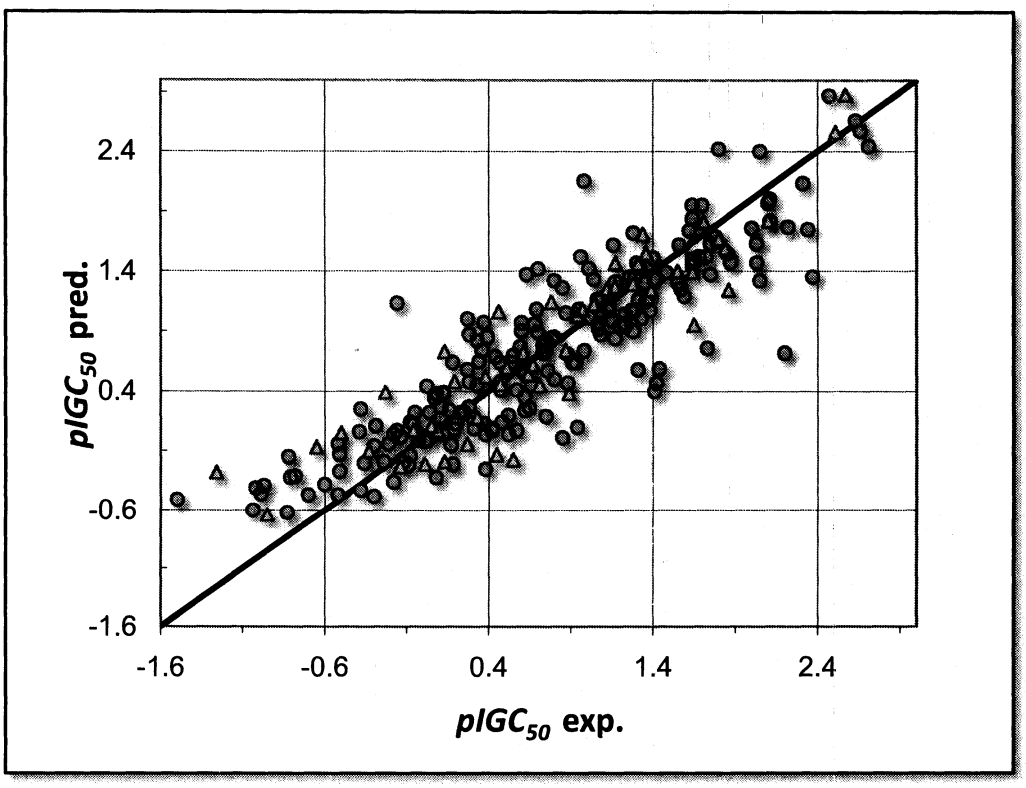

Figura 6.2.3 Valores predichos por Ec. (6.2.4) versus experimentales de pIGC_{50}

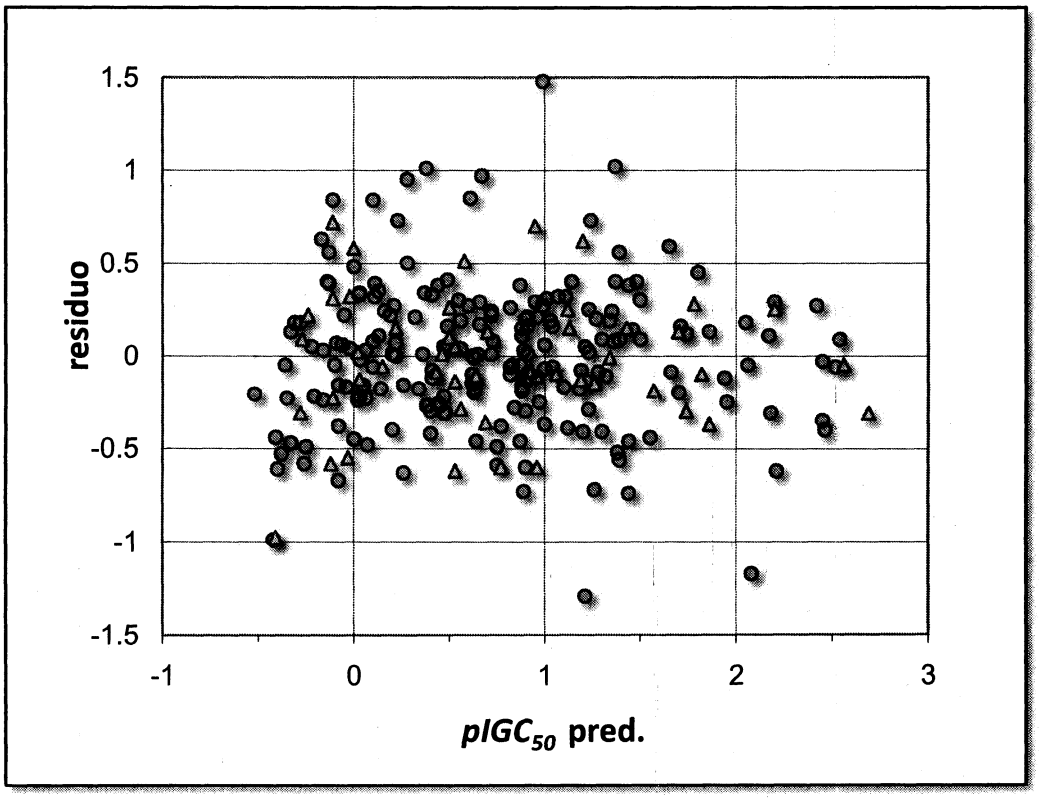

Figura 6.2.4 Gráfico de la dispersión de residuos para la Ec. (6.2.4).

6.3 Predicción QSAR de la inhibición de la aldosa reductasa por flavonoides

6.3.1 Introducción

Los pacientes diabéticos normalmente sufren complicaciones como cataratas, neuropatías periféricas y enfermedades vasculares en particular de la retina, como así de los riñones y corazón. La actividad incrementada de la ruta del sorbitol del metabolismo de la glucosa ha sido involucrada con la patogénesis de estas complicaciones.[129]

La ruta de sorbitol contiene dos enzimas, la Aldosa Reductasa (AR) (EC 1.1.1.21) y la Sorbitol Deshidrogenasa (EC 1.1.1.14).

AR normalmente reduce la glucosa a sorbitol usando Nicotiamida-Adenina Dinucleótido fosfato (NADPH) como cofactor; al mismo tiempo la Sorbitol Deshidrogenasa oxida sorbitol a fructosa. Sin embargo, en condiciones de diabetes, los niveles de glucosa en esta ruta se ven incrementados y por lo tanto se produce sorbitol más rápido de lo que se oxida a fructosa.[130]

La acumulación de sorbitol en el cristalino, nervio o retina resulta en un efecto hiperosmótico que lleva a una inflamación del cristalino y subsecuente formación de cataratas como así también cambios patológicos en otros tejidos.[131] La inhibición de AR es un posible tratamiento y prevención de estos efectos negativos. [132]

Diversos flavonoides y derivados han aparecido en publicaciones indicando poseer actividad inhibitoria contra la enzima AR.[133-135]

Los flavonoides (fenil-benzopiranos) son productos extraídos de las plantas con bajo peso molecular, estos son abundantes, relativamente simples de sintetizar y presentan varias actividades biológicas de interés, consecuentemente su estudio es de un gran interés en muchos campos de investigación.

Claramente es de gran interés poder predecir el valor IC_{50} de compuestos que no tengan datos experimentales, cómo así intentar determinar los parámetros estructurales de los que depende la inhibición de la AR. Una estrategia ampliamente aceptada para sobrellevar la falta de medidas experimentales en sistemas biológicos complejos es el análisis basado en QSAR.[3]

Un estudio QSAR reciente sobre un conjunto de 75 datos de actividades inhibitorias en la enzima AR por flavonoides fue publicado usando regresiones lineales múltiples con descriptores clásicos y cuánticos.[136] El modelo encontrado carecía de significancia estadística mostrando bajos coeficientes de correlación y habilidad predictiva. Un segundo estudio usó el mismo conjunto de datos para un análisis lineal múltiple seleccionando los modelos mediante Algoritmos Genéticos (AG), seguidos de ANN para mejorar el poder predictivo de los modelos lineales, sin embargo las mejoras no fueron significativas.[137]

En este trabajo investigamos un modelo QSAR para la inhibición de AR que puede servir como una guía para el diseño racional de otros inhibidores más potentes y selectivos que contengan el esqueleto de una flavona (Figura 6.3.1) o cromona (Figura 6.3.2); esta última básicamente es una flavona que no contiene el grupo fenilo en la posición 2.

Un gran número de descriptores moleculares que incluyen definiciones de todas las clases fueron analizados para buscar el subconjunto de variables óptimas, usando RM[77-79, 116] para luego refinar los resultados usando el más novedoso ERM.[82] Los resultados fueron comparados con los de los ampliamente aplicados AG.[75] El principal interés es aplicar el nuevo modelo QSAR para estimar la actividad de un grupo de flavonoides recientemente sintetizados,[138] ya que estos no poseen datos experimentales de la inhibición de la AR.

6.3.2 Métodos

6.3.2.1 Conjuntos de datos

En este trabajo usamos un conjunto de calibración de 56 derivados flavonoides para los cuales la actividad estaba informada en la literatura por Štefanič-Petek et al.[136] Inicialmente se intentó usar todo el conjunto de 75 moléculas de este trabajo encontrando una muy pobre correlación entre la estructura y la actividad. Al intentar solucionar esto, lo cual significó un trabajo arduo que llevó a revisar las referencias que contenían las medidas experimentales de forma de clarificar algunas dudas sobre las estructuras,[134, 139] se encontraran importantes errores en la representación de las

estructuras en el trabajo de Štefanič-Petek. Se corrigieron todas aquellas estructuras de las cuales se tenía certeza quedando un conjunto de calibración de 56 derivados de la flavona. Más precisamente esta reducción fue llevada a cabo porque algunas moléculas directamente no presentaban el esqueleto flavonoide deseado o porque su estructura no fue encontrada en las referencias.

Los datos experimentales de efectos inhibitorios sobre AR de las moléculas elegidas fueron medidos espectrofotométricamente; la reacción fue iniciada por la adición de los flavonoides y la velocidad de oxidación de NADPH se determinó siguiendo el descenso de la absorbancia 340 nm. La AR fue obtenida del drenado del cristalino de ojos de ratas de Wistar pesando 200-250g[134, 139] y purificado de acuerdo con el método de Inagaki et al.[140] El valor IC_{50} significa la concentración micro molar del compuesto necesaria para una inhibición del 50% de la enzima y fue determinado por el método de Kador et al.[141]

En adición al conjunto antes mencionado, un conjunto de 4 flavonas con el esqueleto deseado,[142] fue agregado para validar la habilidad predictiva del nuevo modelo encontrado. En este caso las condiciones experimentales fueron casi idénticas a las condiciones del conjunto de calibración, con la diferencia que la AR fue obtenida del cristalino de ojos de ratas Sprague-Dawley de peso 250-280g.[142] Es esperable que esta diferencia no afecte de manera significativa las medidas experimentales.

La Tabla 6.3.1 resume las estructuras moleculares, numero de los sustituyentes y valores experimentales de $-\log IC_{50}$ de los mencionados derivados de la flavona.

6.3.2.2 Descriptores Moleculares

Las moléculas fueron pre optimizadas con un método de *Molecular Mechanics Force Field* (MM+), y luego se refinó la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere y un límite de gradiente de 0.01 kcal.Å-1

Se usó el software Dragon[119] para calcular los descriptores moleculares a los cuales se les adicionaron 18 descriptores constitucionales que tiene en cuenta grupos funcionales y su posición en la molécula; y cuatro derivados de la química-cuántica (momento dipolar molecular, energías totales y energías de HOMO-LUMO) no

incluidos en el software antes mencionado, resultando un grupo de D=1233 descriptores moleculares de distintas clases[120].

Se usaron validaciones cruzadas *l-o-o* y *l-n%-o*[36], con *n%=30%* (16 flavonoides) siendo este el número de moléculas removidas del grupo de calibración y 5000000 el número de casos aleatorios empleados.

6.3.2.3 Búsqueda del mejor modelo

En los cálculos se empleó el sistema de computación Matlab 5.0[128] usando el método de búsqueda RM[77], el cual consume menos tiempo de cálculo para determinar el número de variables optimas d_{opt} y luego se usó el más novedoso ERM[82] para encontrar el mejor modelo de d_{opt} descriptores. Adicionalmente se llevó a cabo una búsqueda usando GA[75] para comparar los resultados obtenidos.

6.3.3 Resultados y Discusión

Inicialmente se establecieron distintas relaciones que vinculan la estructura molecular con la actividad inhibitoria de los flavonoides por medio de regresiones lineales con modelos de 1 a 10 parámetros (d) que fueron seleccionados de el conjunto total de 1233 (D) descriptores. La aplicación del RM al conjunto de calibración de 56 moléculas mostró que la molécula **53** era un *outlier* en la mayoría de los modelos encontrados. Más precisamente, fue la molécula con mayor error en 110 de 143 modelos probados, en el mejor modelo de seis parámetros la molécula tubo un error de 2.97S. Inesperadamente, la estructura de esta molécula no presenta diferencias significativas con el resto de las moléculas del conjunto de validación. Debido a que la predicción mediante un modelo QSAR no puede ser intrínsecamente mejor que los datos experimentales usados para desarrollar el modelo, la calidad de los datos ingresados tendrá una importante influencia en el desempeño del modelo QSAR,[143] por lo tanto la molécula **53** fue quitada del conjunto de calibración.

Nuevamente se usó RM en el conjunto de calibración resultante de 55 moléculas para calcular los mejores modelos QSAR con $1 \leq d \leq 13$. La Figura 6.3.3 muestra que el máximo de la función *FIT* aparece en d_{max} = 12 del cual se puede determinar que el

número optimo de descriptores es d_{opt} = 6 (Los detalles de como se determian d_{opt} a partir de d_{max} se pueden ver en la sección 4.7). Más precisamente el mejor modelo provisto por RM es:

$$\begin{aligned}-\log \mathrm{IC}_{50} = {} & 4.8501(\pm 1.7) + 12.773(\pm 1.4)\, BELp4 - 4.950(\pm 0.7)\, GGI8 \\ & -12.191(\pm 0.9)\, MATS4e + 0.905(\pm 0.2)\, Mor22e - 16.422(\pm 2.1)\, E1p \\ & -16.844(\pm 1.7)\, R4v\end{aligned} \quad (6.3.1)$$

$$N = 55,\ R = 0.9362,\ S = 0.4364,\ FIT = 3.744,\ p < 10^{-4}$$
$$R_{loo} = 0.914,\ S_{loo} = 0.507,\ R_{l-30\%-o} = 0.763,\ S_{l-30\%-o} = 0.891$$
$$RMSE_{Test\ Set} = 3.9059$$

donde, los errores absolutos de la regresión se encuentran en paréntesis, p es la significancia del modelo y $RMSE_{\text{Test Set}}$ es la raíz cuadrada de los errores medios del conjunto de validación.

Luego usamos ERM[82] para buscar un mejor modelo de d_{opt} = 6 descriptores que en este caso es:

$$\begin{aligned}-\log \mathrm{IC}_{50} = {} & -85.5375(\pm 10) - 10.882(\pm 1.4)\, E1u - 15.398(\pm 0.9)\, MATS4e \\ & +55.920(\pm 5.35)\, BELm2 + 7.7606(\pm 0.9)\, HATS6e - 2.6755(\pm 0.5)\, DISPe \\ & -18.253(\pm 1.1)\, R4p\end{aligned} \quad (6.3.2)$$

$$N = 55,\ R = 0.9523,\ S = 0.3789,\ FIT = 5.14,\ p < 10^{-5}$$
$$R_{loo} = 0.934,\ S_{loo} = 0.447,\ R_{l-30\%-o} = 0.803,\ S_{l-30\%-o} = 0.886$$
$$RMSE_{Test\ Set} = 2.9127$$

Como punto de referencia dejamos que un AG seleccione un modelo optimo de d_{opt} = 6 descriptores. Para esto se optimizaron los parámetros del AG para este problema en particular llevando a cabo numerosas pruebas encontrando los siguientes ajustes: Número de individuos = 250; Brecha Generacional = 0.9; Probabilidad de entrecruzamiento = 0.6; Probabilidad de mutación = 0.7/*d*. El criterio para frenar la evolución fue determinado de forma tal que cuando un individuo ocupara más del 90% de la población o cuando el número de generaciones fuese 2500 se interrumpiría el proceso. El modelo óptimo AG encontrado es:

$$-\log IC_{50} = 12.2967(\pm 1.5) - 0.1898(\pm 0.02)\,TIC0 - 15.6392(\pm 1)\,MATS4e + 1.7611(\pm 0.4)\,H7e - 10.0425(\pm 1.6)\,E1u + 16.6513(\pm 1.9)\,BELe4 - 14.0207(\pm 1.5)\,R3v \qquad (6.3.3)$$

$$N = 55,\ R = 0.9374,\ S = 0.4325,\ FIT = 3.822,\ p < 10^{-4}$$
$$R_{loo} = 0.917,\ S_{loo} = 0.499,\ R_{l-30\%-o} = 0.7739,\ S_{l-30\%-o} = 1.106$$
$$RMSE_{Test\ Set} = 4.1607$$

Al examinar los parámetros estadísticos calculados para los conjuntos de calibración y validación se llegó a la conclusión que ERM produce mejores resultados que AG y RM.

La Tabla 6.3.2 muestra un resumen de los modelos lineales con 1 a d_{opt} + 1 parámetros para RM y d_{opt} parámetros para ERM y AG. Los detalles de los descriptores moleculares de la Tabla 6.3.2 se presentan en la Tabla 6.3.3.

Se utilizó el método de la variable Y aleatoria[105] para demostrar que los resultados obtenidos en Ec. (6.3.2) no eran fortuitos (para más detalles sobre el método ver sección 5.3.2). Luego de analizar 1000000 de casos el menor valor obtenido es S = 0.8254 siendo este considerablemente mayor al obtenido de la calibración S = 0.3789. Esto sugiere que el modelo encontrado es robusto y que la relación estructura actividad es confiable.

El grafico de valores predichos vs. experimentales de $-\log IC_{50}$ mostrado en la Figura 6.3.4 sugiere que las 55 flavonas siguen una línea recta. La Tabla 6.3.1 muestra las potencias inhibitorias predichas por Ec.(6.3.2) para el conjunto de calibración y validación. El comportamiento de los residuos en relación a los datos experimentales ilustrados en la Figura 6.3.5 muestra una distribución normal para los dos conjuntos. Esta figura omite la molécula **59** la cual tiene un residuo mayor a $3S$ =1.14. Esta desviación puede ser un defecto estadístico del modelo o una consecuencia de características físicas. Aunque no es posible responder a este interrogante mediante el presente análisis QSAR, cabe mencionar que esta molécula presenta un residuo del mismo orden y signo en el resto de los modelos usados para determinar d_{opt}, lo que sugiere que pude existir un error en el dato experimental correspondiente.

La matriz de correlación de la Tabla 6.3.4 muestra que los descriptores no están inter correlacionados de manera importante (R_{ij} < 0.599), lo cual justifica el uso de todos los parámetros en la ecuación. El poder predictivo del modelo es satisfactorio cuando es probado frente a la inclusión y exclusión de compuestos, esto fue medido con los parámetros estadístico de R_{loo} = 0.934 y $R_{l\text{-}30\%\text{-}o}$ = 0.803. De acuerdo a la literatura

$R_{l\text{-}n\%\text{-}o}$ debe ser mayor que 0.71 para poder considerar un modelo como correctamente validado.[104]

La estandarización de los coeficientes[74] de la regresión de la Ec.(6.3.2) permite asignar una importancia mayor a los descriptores que muestran un coeficiente estandarizado absoluto mayor (mostrado entre paréntesis):

$$MATS4e(1.11) > R4p(0.83) > BELm2\ (0.62) > HATS6e(0.59) > Elu(0.37) > DISPe(0.24) \quad (6.3.4)$$

El ordenamiento dado por la Ec. (6.3.4) muestra que la auto correlación bidimensional *MATS4e* y el descriptor de tipo GETAWAY llamado *R4p* son las variables más relevantes para el presente grupo de flavonoides. El descriptor *MATS4e* indica que la actividad puede tener una importante dependencia con la electronegatividad de los átomos que forman la molécula. El descriptor de tipo 3D más relevante es el *R4p* es de esperar que el mismo tenga una gran dependencia frente a cambios conformacionales dado que codifica información de pares de átomos que se encuentran considerablemente lejos unos de otros (distancia de 4). Por esta razón es posible decir que actividad inhibitoria para el presente conjunto tenga una importante dependencia frente a cambios conformacionales.

Por medio de la Ec. (6.3.2) se calcularon la actividad inhibitoria -logIC_{50} frente a AR de los derivados nuevos. Los resultados se encuentran en la Tabla 6.3.5. Los cálculos sugieren que aquellos flavonoides con un grupo naftilo probablemente presenten una alta actividad y son consecuentemente candidatos para estudios posteriores. Por el otro lado las moléculas con un grupo furanilo tendrían una muy baja actividad y pueden en principio ser descartados como candidatos.

6.3.4 Conclusiones

Se logró construir un modelo QSAR con buen carácter predictivo de la actividad inhibitoria sobre la enzima AR para 55 flavonoides para los cuales no se había logrado llevar a cabo un estudio satisfactorio; siendo esta actividad de muy alta importancia en el tratamiento de pacientes con diabetes. Se usaron seis descriptores moleculares que tienen en cuenta aspectos bi y tridimensionales. Usando el nuevo modelo se logró

predecir la actividad de flavonoides recientemente sintetizados que no poseían datos experimentales, mostrando que la presencia de grupos funcionales naftilo aumenta la actividad inhibitoria mientras que los grupos furanilos la disminuyen de manera significativa.

Los importantes resultados obtenidos en el trabajo tuvieron como consecuencia su publicación en una importante revista: Mercader, A. G.; Duchowicz, P. R.; Fernandez, F. M.; Castro, E. A. Daniel O. Bennardi, D. O., Autino, J. C. Romanelli, G. P., Bioorganic & Medicinal Chemistry, 16 (2008) 7470-7476. Es de esperar que este aporte a la comunidad científica ayude a encontrar nuevos flavonoides con valores más altos de la actividad estudiada.

Tabla 6.3.1. Valores experimentales y calculados (Ec. (6.3.2)) $-\log IC_{50}$

NOTA: Los sustituyentes se basan en un esqueleto de flavona (Figura 6.3.1)

Nº	Sustituyentes	$-\log IC_{50}$ Exp.	$-\log IC_{50}$ Pred.
	Conjunto de Calibración		
1	5,7,3',4'-OH; 3,6-OCH_3	7.553	7.374
2	3',4'-OH; 5,6,7,8-OCH_3	7.490	6.922
3	6,3',4'-OH; 5,7,8-OCH_3	7.456	7.170
4	5,7,3',4'-OH; 6-OCH_3; 8-CH_2Ph	7.470	7.654
5	5,3',4'-OH; 6,7,8-OCH_3	7.410	7.014
6	3',4'-OH; 5,7,8-OCH_3	7.350	6.568
7	5,6,7,3',4'-OH; 3-OCH_3	7.240	7.518
8	5,6,3',4'-OH; 7,8-OCH_3	7.190	7.333
9	7,3',4'-OH; 5,8-OCH_3	7.130	7.078
10	5,3',4'-OH; 7,8-OCH_3	7.110	7.117
11	3',4'-OH; 5,6,7-OCH_3	7.040	7.187
12	5,6,7,3',4'-OH; 8-OCH_3	6.920	6.585
13	6,3',4'-OH; 5,7-OCH_3	6.850	6.509
14	4'-OH; 5,6,7,8-OCH_3	6.796	6.558

15	8,3',4'-OH; 5,7-OCH_3	6.790	6.508
16	3',4'-OH; 3,5,7,8-OCH_3	6.770	6.689
17	5,6,7,3',4'-OH	6.690	6.598
18	5,3',4'-OH; 6,7-OCH_3	6.770	6.995
19	5,8,3',4'-OH; 7-OCH_3	6.640	7.167
20	5,7,3',4'-OH; 3,8-OCH_3	6.620	6.697
21	6,4'-OH; 5,7,8-OCH_3	6.600	6.631
22	3',4'-OH; 3,5,6,7-OCH_3	6.570	6.853
23	5,7,3',4'-OH; 8-OCH_3	6.550	6.667
24	7,3',4'-OH; 3,5,8-OCH_3	6.550	6.367
25	8-OCH_3; 5,6,7,3',4'-$OCOCH_3$	6.520	6.336
26	5,6,3',4'-OH; 7-OCH_3	6.520	6.467
27	6,3',4'-OH; 3,5,7-OCH_3	6.520	6.830
28	5,3',4'-OH; 3,6,7-OCH_3	6.458	6.267
29	5,7,4'-OH; 6,8-OCH_3	6.390	6.402
30	5,4'-OH; 6,7,8-OCH_3	6.270	6.394
31	5,6,3',4'-OH; 3,7-OCH_3	6.090	6.668
32	5,6,4'-OH; 7,8-OCH_3	6.070	6.679
33	5,6,7,4'-OH; 8-OCH_3	5.920	5.782
34	5,6,7,4'-OH; 8,3'-OCH_3	5.920	5.207
35	5,4'-OH; 6,7-OCH_3	5.850	5.475
36	5,7,3',4'-OH; 3-O-Rh	5.933	5.966
37	5,7,4'-OH; 6,8,3'-OCH_3	5.350	5.276
38	6,4'-OH; 5,7,8,3'-OCH_3	5.200	5.118
39	5,4'-OH; 6,7,3'-OCH_3	5.170	5.284
40	5,7-OH; 6,8,4'-OCH_3	5.140	4.824
41	5,6,7-OH; 8-OCH_3	5.090	4.964

42	5,6-OH; 7,8-OCH_3	5.076	5.155
43	3',4'-OH; 5,6,7-OCH_3; 3-$COCH_3$	5.050	4.581
44	5,3'-OH; 6,7-OCH_3; 4'-O-Glc	5.086	4.689
45	5-OH; 6,7,3'-OCH_3; 4'-O-Glc	4.880	4.900
46	5-OH; 6,7-OCH_3; 4'-O-Glc	4.790	4.477
47	5,7-OH; 6,8,3'-OCH_3; 4'-O-Glc	4.740	4.521
48	4'-OH; 5,6,7,8,3'-OCH_3	4.730	5.406
49	5,4'-OH; 6,8,3'-OCH_3; 7-O-Glc	4.680	5.185
50	5,7-OH; 6,8,3',4'-OCH_3	4.530	4.783
51	5,4'-OH; 6,7,8,3'-OCH_3	4.340	5.323
52	5,6,4'-OH; 7,8,3'-OCH_3	3.960	4.748
53	6-OH; 5,7,8-OCH3	3.540	-
54	5,5'-OH; 7,2',4'-OCH_3	3.500	3.591
55	7-OH; 5-OCH_3	3.000	2.977
56	5,4'-OH; 7,2',5'-OCH_3	3.000	3.291
	Conjunto de Validación		
57	7-OH; 2'-OH	5.780	6.206
58	7-OH; 2' 4'-OH	5.640	5.254
59	6-OH; 4'-OH	5.280	10.33
60	7-OH; 2',4'-OH	6.456	5.592

Tabla 6.3.2 Modelos lineales QSAR para el conjunto de calibración de $-\log IC_{50}$ (N=55). La mejor relación aparece en negrita.

Modelo	**DESCRIPTORES USADOS**	***R***	***S***	***FIT***
M1	*LUMO*	0.616	0.931	0.579
M2	*DISPp, C-027*	0.739	0.804	1.059
M3	*Mor32m, H-048, >0.2*	0.826	0.679	1.715

M4	*MATS4e, E1u, HATS6e, R4m*	0.878	0.582	2.379
M5	*GATS4e, DISPe, E1u, HATS5m, R4m*	0.900	0.536	2.607
M6	*BELp4, GGI8, MATS4e, Mor22e, E1p, R4v* (Ec.(6.3.1))	0.936	0.436	3.744
M7	*SPP, DISPe, RDF140m, E1p, H4m, Dipole Moment, LUMO*	0.950	0.392	4.181
M6B	***E1u, MATS4e, BELm2, HATS6e, DISPe, R4p* (Ec. (6.3.2))**	**0.952**	**0.379**	**5.140**
M6C	*TIC0, MATS4e, H7e, E1u, BELe4, R3v* (Ec. (6.3.3))	0.937	0.433	3.822

Tabla 6.3.3 Símbolos para los descriptores moleculares usados en los diferentes modelos.

Descriptor	Tipo	Definición
LUMO	Quantum-Chemical	Lowest Unoccupied Molecular Orbital energy (eV)
DISPp	Geometrical	d COMMA2 value / weighted by atomic polarizabilities.
C-027	Atom-Centred Fragments	*C-027* corresponds to: R—CH—X
Mor32m	3D-MoRSE	3D-MoRSE - signal 32 / weighted by atomic masses
H-048	Atom-Centred Fragments	H attached to C2(sp3) / C1(sp2) / C0(sp).
>0.2	Charge	Number of atoms with charge higher than 0.2
MATS4e	2D Autocorrelations	Moran autocorrelation - lag 4 / weighted by atomic Sanderson electronegativities.
E1u	WHIM	1st component accessibility directional WHIM index / unweighted.
HATS6e	GETAWAY	Leverage-weighted autocorrelation of lag 6 / weighted by atomic Sanderson electronegativities.
R4m	GETAWAY	R autocorrelation of lag 4 / weighted by atomic masses.
GATS4e	2D Autocorrelations	Geary autocorrelation - lag 4 / weighted by atomic Sanderson electronegativities.
DISPe	Geometrical	d COMMA2 value / weighted by atomic Sanderson electronegativities.
HATS5m	GETAWAY	leverage-weighted autocorrelation of lag 5 / weighted by atomic masses.

BELp4	BCUT	Lowest eigenvalue n. 4 of Burden matrix / weighted by atomic polarizabilities.
GGI8	Topological	Topological charge index of order 8
Mor22e	3D-MoRSE	3D-MoRSE - signal 22 / weighted by atomic Sanderson electronegativities.
E1p	WHIM	1st component accessibility directional WHIM index / weighted by atomic polarizabilities.
R4v	GETAWAY	R autocorrelation of lag 4 / weighted by atomic van der Waals volumes.
SPP	Charge	Subpolarity parameter
RDF140m	Radial Distribution Function	Radial Distribution Function - 14.0 / weighted by atomic masses.
H4m	GETAWAY	H autocorrelation of lag 4 / weighted by atomic masses.
Dipole Moment	Quantum-Chemical	Total Molecular Dipole Moment (Debyes)
BELm2	BCUT	lowest eigenvalue n. 2 of Burden matrix / weighted by atomic masses.
R4p	GETAWAY	R autocorrelation of lag 4 / weighted by atomic polarizabilities.
TIC0	Topological	total information content index (neighborhood symmetry of 0-order).
H7e	GETAWAY	H autocorrelation of lag 7 / weighted by atomic Sanderson electronegativities.
BELe4	BCUT	lowest eigenvalue n. 4 of Burden matrix / weighted by atomic Sanderson electronegativities.
R3v	GETAWAY	R autocorrelation of lag 3 / weighted by atomic van der Waals volumes.

Tabla 6.3.4 Matriz de correlación para los descriptores de la Ec. (6.3.2) (N=55).

	Elu	MATS4e	BELm2	HATS6e	DISPe	R4p
Elu	1	0.1226	0.2569	0.3177	0.0431	0.1928
MATS4e		1	0.2742	0.3455	0.0274	0.0674
BELm2			1	0.0482	0.3041	0.5992
HATS6e				1	0.0137	0.0578
DISPe					1	0.259
R4p						1

Tabla 6.3.5 Datos calculados de $-\log IC_{50}$ (Ec. (6.3.2)) para las nuevas moléculas.

NOTA: Los sustituyentes de las estructuras 61 a 63 se basan en un esqueleto de flavona (Figura 6.3.1) y los de las estructuras 64 a76 en un esqueleto de cromona (Figura 6.3.2)

Nº	Sustituyentes	$-\log IC_{50}$ Pred.
Conjunto de predicción (flavonas, Figura 6.3.1)		
61	7-OCH3	7.566
62	7-Cl	6.873
63	7-Br	6.570
Conjunto de predicción (cromonas, Figura 6.3.2)		
64	2-(2-furyl)	-2.184
65	2-(β-naphtyl)	11.064
66	2-(α-naphtyl)	9.455
67	7-Br, 2-(β-naphtyl)	7.031
68	7-Cl, 2-(α-naphtyl)	5.881
69	7-CH_3, 2-(α-naphtyl)	8.828
70	7-Br, 2-(α-naphtyl)	5.405
71	7-OCH_3, 2-(β-naphtyl)	8.440

72	7-OCH_3, 2-(α-naphtyl)	6.259
73	7-Cl, 2-(β-naphtyl)	7.415
74	7-Cl, 2-(2-furyl)	-4.220
75	7-F, 2-(α-naphtyl)	6.883
76	7-CH_3, 2-(β-naphtyl)	10.415

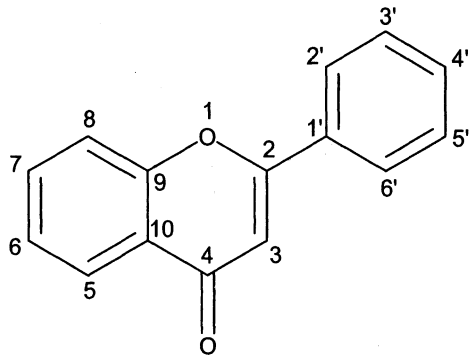

Figura 6.3.1 Estructura molecular de la flavona.

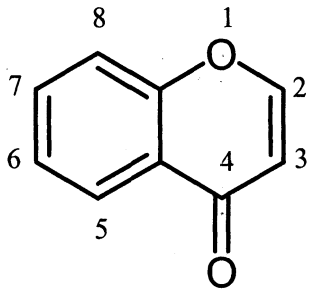

Figura 6.3.2 Estructura molecular de la cromona.

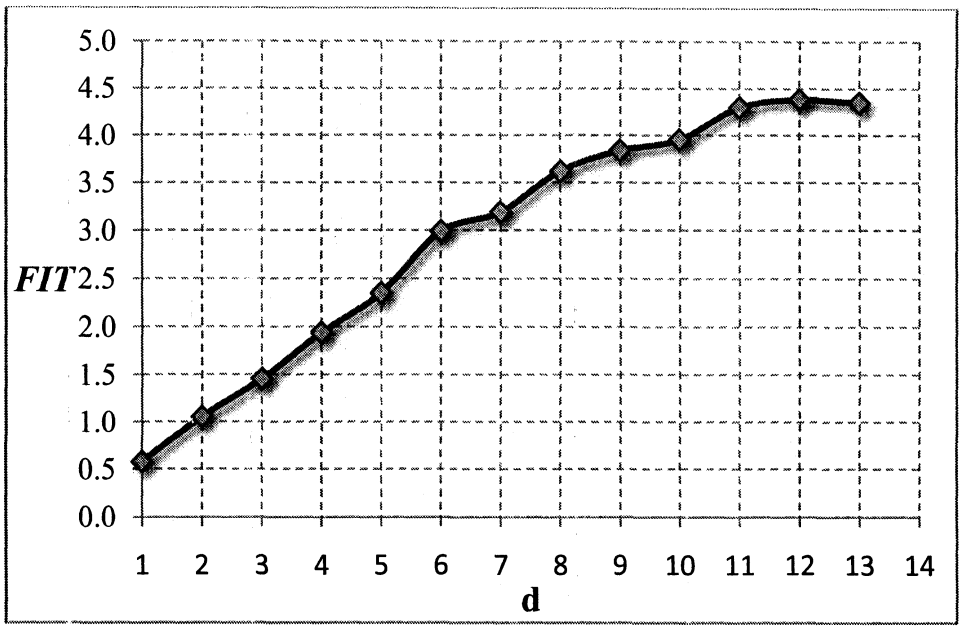

Figura 6.3.3 Parámetro *FIT* vs número de descriptores para el conjunto de calibración.

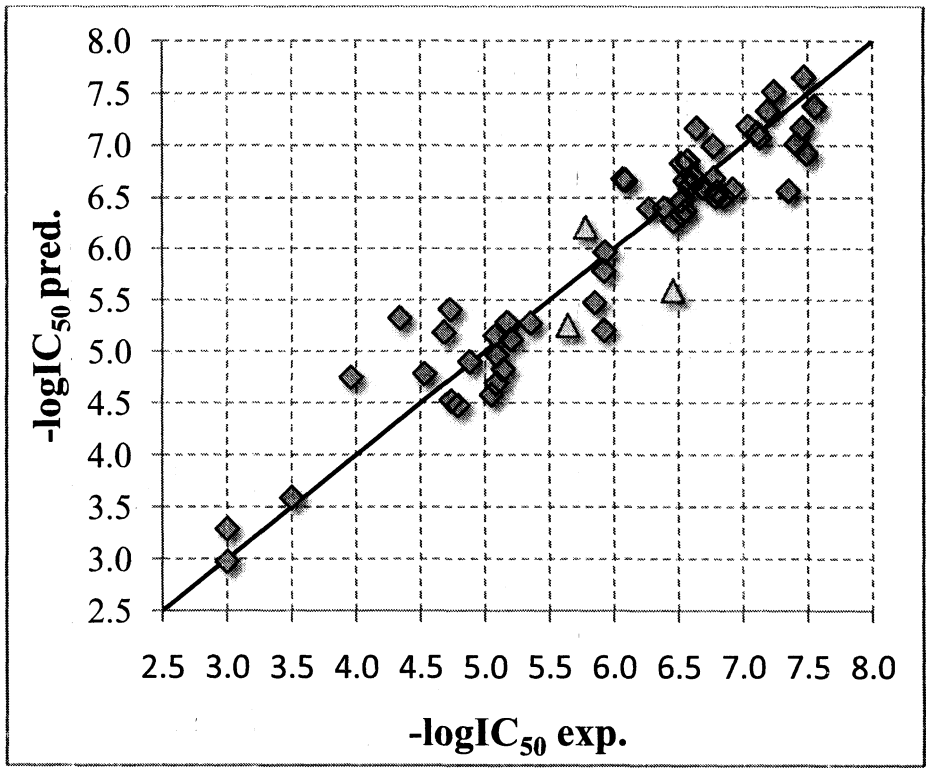

Figura 6.3.4 Valores predichos por la Ec. (6.3.2) versus experimentales de $-\log IC_{50}$ para el conjunto de calibración (rombos) y validación (triángulos).

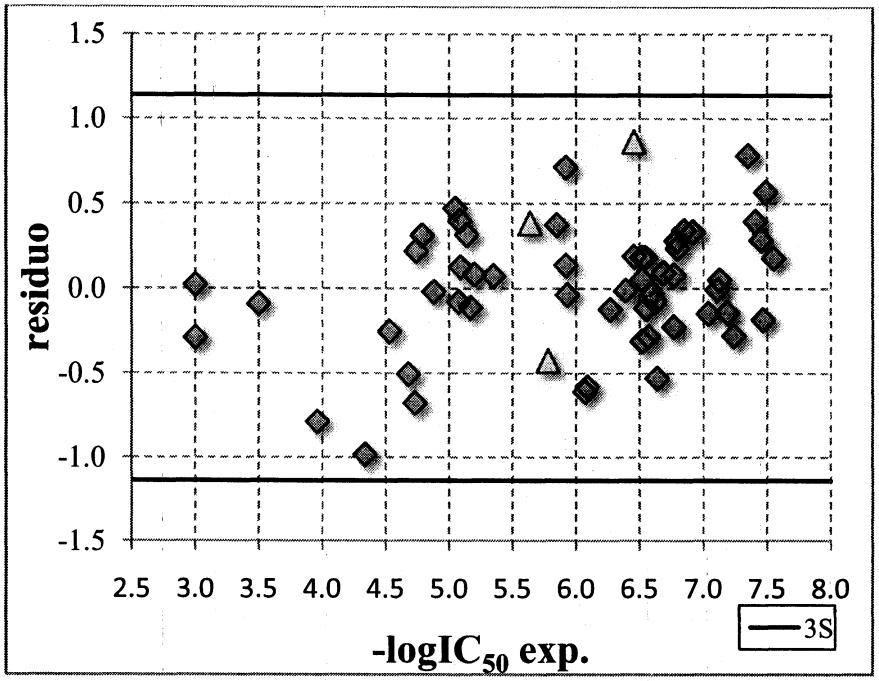

Figura 6.3.5 Gráfico de la dispersión de los residuos para los conjuntos de calibración y validación usando Ec. (6.3.2).

6.4 Estudio QSPR del efecto del solvente en la desactivación de la emisión $^5D_0 \rightarrow {}^7F_2$ del Eu(6,6,7,7,8,8,8-heptafluoro-2,2-dimetil-3,5-octanedionato)$_3$

6.4.1 Introducción

Los quelatos de lantánidos han atraído un gran interés debido a que pueden ser usados en diversas aplicaciones como agentes de extracción de solventes, reactivos en RMN y en materiales laser.[144] Por sus propiedades lumínicas únicas, sus aplicaciones se están expandiendo en una gran variedad de ensayos bio analíticos, en diagnostico, investigación, desarrollo de nuevas de drogas, como herramientas en sensores y en imágenes. El desarrollo de quelatos altamente lumínicos, ligandos quelatantes y varias formas de nano lechos de lantánidos lumínicos, ha hecho posible construir un gran número de tecnologías de ensayos permitiendo no solo una alta sensibilidad mediante la

discriminación del fondo, sino también ensayos muy simples y robustos adecuados para análisis clínicos de alto caudal y miniaturización.

La propiedades espectrales y temporales únicas de los lantánidos también permiten ensayos múltiples para examinar varias muestras simultáneamente o para extraer más datos de una única muestra. El Terbio y el Europio pueden tener tiempos de vida de milisegundos y espectros de emisión con picos definidos aún en soluciones acuosas. Consecuentemente esos lantánidos son de interés considerable en medidas de fluorescencia resueltas en el tiempo, en particular para discernir frente a fluorescencia de fondo que tenga mayor velocidad (nanosegundos). Han mostrado además ser excelentes dadores en medidas de transferencia de energía por luminiscencia, para medir distancias entre los 20–100 Å, entre y dentro de macromoléculas biológicas.[145-148]

Se ha investigado con anterioridad la desactivación de la luminiscencia por solventes y temperatura de Eu(6,6,7,7,8,8,8-heptafluoro-2,2-dimetil-3,5-octanedionato)$_3$ o Eu(fod)$_3$.[149, 150] La habilidad de las vibraciones de alta energía del O-H para desactivar los estados excitados f-f es conocida. Sin embargo, otros efectos específicos en la desactivación de la luminiscencia en solventes con enlaces C-H y/o C-X (X=haluro) permanece sin explicación. El tiempo de vida de la luminiscencia (τ) del Eu(fod)$_3$ en diferentes solventes es un parámetro fundamental en el intento de explicar el mecanismo de desactivación.[150]

Por lo antes mencionado queda claro que es de gran interés poder predecir esta propiedad en solventes para los que no haya medidas experimentales, como así intentar determinar cuales parámetros estructurales del solvente tienen efecto en τ. Una manera de llevar a cabo esto es mediante un análisis QSPR.

En el presente trabajo se predicen los tiempos de vida de la luminiscencia de Eu(fod)$_3$ para 25 solventes distintos cuyos datos experimentales fueron recolectados de la literatura.[150]

Un gran número de descriptores moleculares que incluían definiciones de todas las clases se exploró usando el recientemente desarrollado ERM[82] Los resultados se compararon con los anteriormente aplicados RM[77-79, 116] y AG. [76]

6.4.2 Métodos

6.4.2.1 Datos

Se eligió un conjunto de calibración de 23 solventes y uno de validación de 2 solventes para los cuales τ fue medido en nuestro instituto. El tamaño del conjunto de calibración fue elegido de forma de a tener la mayor cantidad de información estructural en el modelo calculado. Todos los tiempos de vida fueron medidos a temperatura ambiente usando pulsos de un laser de nitrógeno con excitación a 337 nm y examinando la señal a longitudes de onda entre 560 y 750 nm para el compuesto de Eu (III).[150] La Tabla 6.4.1 muestra los nombres de los solventes usados junto con los datos experimentales de τ.

6.4.2.2 Descriptores Moleculares

Las moléculas fueron pre optimizadas con un método de *Molecular Mechanics Force Field* (MM+), y luego se refinó la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere y un límite de gradiente de 0.01 kcal.Å-1

Se usó el software Dragon[119] para calcular los descriptores moleculares a los cuales se les adicionaron 18 descriptores constitucionales que tiene en cuenta grupos funcionales y su posición en la molécula; y cuatro derivados de la química-cuántica (momento dipolar molecular, energías totales y energías de HOMO-LUMO) no incluidos en el software antes mencionado, resultando un grupo de D=1057 descriptores moleculares de distintas clases.[120]

Se usaron validaciones cruzadas *l-o-o* y *l-n%-o*[36], con *n%=22%* (5 solventes) siendo este el número de moléculas removidas del grupo de calibración y 5000000 fue el número de casos aleatorios empleados.

6.4.2.3 Búsqueda del mejor modelo

En los cálculos se empleó el sistema de computación Matlab 5.0[128] usando el novedoso método de búsqueda ERM[82] para encontrar el mejor modelo de d_{opt} descriptores, luego se usaron RM[77] y GA[75] para comparar los resultados obtenidos.

6.4.3 Resultados y Discusión

De forma de determinar el número optimo de descriptores se probaron diferentes relaciones predictivas con la habilidad de vincular la estructura molecular del solvente con el tiempo de vida de luminiscencia (τ) del $Eu(fod)_3$, por medio de modelos de regresiones lineales de 1 a 14 parámetros (d) seleccionados por ERM del conjunto total de D=1057 descriptores.

Para determinar el número óptimo de parámetros se empleó el novedoso método que hace uso del parámetro *VFIT* antes presentado en la sección 4.7. En este caso cuando k en *VFIT* es incrementado (ver Tabla 6.4.2) un primer máximo aparece en d=9 (k=2), un segundo en d=7 (k=2.5) y finalmente un tercero en a d=4 (k=3) que cumple con la regla que indica que para este grupo de solventes se podría usar un máximo de 5 parámetros.[98]

La función resultante *VFIT* con k=3 aumenta con d hasta un máximo valor de $d=d_{max}=4$ mostrado en la Figura 6.4.1, por lo que se deduce que este es el número de descriptores optimo. La Figura 6.4.1 también muestra que *FIT* no presenta un máximo en el intervalo de d entre 1 y 14, no permitiendo determinar el número optimo de variables. La Tabla 6.4.2 muestra que d_{max}=4 permanece inalterado por dos incrementos adicionales de k lo que respalda el hecho que este es efectivamente el número optimo de descriptores.

Por lo tanto podemos llegar a la conclusión que el mejor modelo QSPR de acuerdo a ERM es:

$$\tau = 0.6903(\pm 0.04) + 0.1575(\pm 0.03)\cdot GATS5v - 0.1731(\pm 0.01)\cdot Sp + 0.2010(\pm 0.03)\cdot Jhetv + 0.2120(\pm 0.01)\cdot RDF015e \quad (6.4.1)$$

$$N = 23,\ R = 0.9746,\ S = 0.0375,\ FIT = 8.7242,\ p < 10^{-4}$$
$$R_{loo} = 0.9597,\ S_{loo} = 0.0470,\ R_{l-22\%-o} = 0.9031,\ S_{l-22\%-o} = 0.0735$$
$$RMSE_{Val} = 0.03808$$

donde, los errores absolutos de la regresión se encuentran en paréntesis, p es la significancia del modelo, *FIT* la función de Kubinyi, *loo* y *l-22%-o* significan validaciones cruzadas de *Leave-One-Out* y *Leave-More-Out*, respectivamente y $RMSE_{Val}$ es la raíz cuadrada de los errores medios del conjunto de validación.

Luego al aplicar RM se obtuvieron exactamente los mismos cuatro descriptores del modelo (6.4.1).

También se probaron los AG en el mismo problema, lo que requirió de numerosos intentos para optimizar los parámetros. Los mejores parámetros encontrados fueron: Número de individuos = 20; Brecha generacional = 0.9; Probabilidad de entrecruzamiento = 0.2; Probabilidad de mutación = 0.7/d. El criterio para frenar la evolución fue determinado de forma tal que cuando un individuo ocupara más del 90% de la población o cuando el número de generaciones llegase a 2500 se interrumpiría el proceso. El mejor modelo encontrado luego de diez corridas usando los parámetros optimizados es:

$$\tau = 0.9826(\pm 0.06) - 0.6898(\pm 0.1)\cdot Mor30e - 0.0098(\pm 0.0007)\cdot piPC05 + 0.1810(\pm 0.03)\cdot BEHm5 - 0.8935(\pm 0.1)\cdot X1Av \qquad (6.4.2)$$

$$N = 23,\ R = 0.968,\ S = 0.042,\ FIT = 6.8713,\ p < 10^{-4}$$
$$R_{loo} = 0.9426,\ S_{loo} = 0.0566,\ R_{l-22\%-o} = 0.8736,\ S_{l-22\%-o} = 0.0977$$
$$RMSE_{Val} = 0.1246$$

Se puede apreciar que ERM/RM presentan mejores parámetros estadísticos que AG. La Tabla 6.4.3 muestra un resumen de los modelos lineales con 1 a d_{opt}+1 parámetros para ERM y d_{opt} parámetros para AG. En esta tabla se puede apreciar que mientras que los parámetros de calibración continúan mejorando, los parámetros estadísticos de validación externa ($RMSE_{Val}$) mejoran al aumentar d hasta llevgar a d_{opt} para luego deteriorarse, esto corrobora la bondad del método usado para seleccionar d_{opt}. La Tabla 6.4.4 muestra el significado de los descriptores de la Tabla 6.4.3.

Se utilizó el método llamado *variable Y aleatoria*[105] para demostrar que los resultados obtenidos en Ec. (6.4.1) no eran fortuitos (para más detalles sobre el método ver sección 5.3.2)). Luego de analizar 5000000 de casos el menor valor obtenido es S=0.06739 siendo este considerablemente mayor al obtenido de la calibración S=0.0375. Esto sugiere que el modelo encontrado es robusto y que la relación estructura actividad es confiable.

El gráfico de valores predichos vs. experimentales para τ mostrado en la Figura 6.4.2 sugiere que los 23 solventes del conjunto da calibración y 2 del de validación aproximadamente siguen una línea recta. La Tabla 6.4.1 también incluye los valores predichos de tiempo de luminiscencia (τ) obtenidos por la Ec. (6.4.1) para los dos conjuntos y los correspondientes residuos. El comportamiento de los residuos en relación a los datos experimentales ilustradas en la Figura 6.4.3 muestra una

distribución normal para los dos conjuntos. Ninguna molécula mostró un residuo mayor a 2S.

La matriz de correlación de la Tabla 6.4.5 muestra que los descriptores no están inter correlacionados de manera importante (R_{ij} < 0.7299),lo cual justifica el uso de todos los parámetros en la ecuación. El poder predictivo del modelo es satisfactorio frente a la inclusión y exclusión de compuestos, esto fue medido con los parámetros estadístico de $R_{loo} = 0.9597$ y $R_{l\text{-}30\%\text{-}o} = 0.9031$. De acuerdo a la literatura $R_{l\text{-}n\%\text{-}o}$ debe ser mayor que 0.71 para poder considerar un modelo como correctamente validado.[104]

De forma de poner el poder predictivo de la Ec. (6.4.1) a una prueba adicional un grupo de 3 solventes fueron seleccionados en base a su relación cualitativa estructura propiedad y fueron agregados al conjunto de validación. Con los restantes 20 solventes se re calibró el modelo llegándose a la siguiente ecuación:

$$\tau = 0.6884(\pm 0.04) + 0.1515(\pm 0.04)\cdot GATS5v - 0.1728(\pm 0.01)\cdot Sp + 0.2007(\pm 0.03)\cdot Jhetv + 0.2137(\pm 0.02)\cdot RDF015e \qquad (6.4.3)$$

$$N = 20,\ R = 0.9737,\ S = 0.0395,\ FIT = 7.5972,\ p < 10^{-4}$$

$$RMSE_{Val} = 0.03247$$

Se puede apreciar que los parámetros de la regresión son similares a los de la Ec. (6.4.1)

Por medio de la Ec. (6.4.3) se predijeron la propiedad de las moléculas del nuevo conjunto de validación y se agregaron los resultados a la Figura 6.4.2 y a la Figura 6.4.3 El gráfico valores predichos vs. valores experimentales de τ sugiere que los 5 solventes del conjunto de validación aproximadamente siguen una línea recta. Nuevamente ninguna molécula puede considerarse como *outlier*.

Con el propósito de determinar si el nuevo conjunto de validación puede ser tomado como un verdadero conjunto externo en la Ec. (6.4.3), una nueva búsqueda mediante ERM se llevó a cabo usando los 20 solventes definidos como conjunto de calibración. El mejor modelo encontrado en este caso resultó casi idéntico al de la Ec. (6.4.1) basado en el conjunto de 23 solventes y muestra similar calidad estadística para ambos conjuntos de calibración y validación. La única diferencia es que el descriptor *Jhetv* se reemplazó con *Jhetp*; estos dos descriptores tienen la misma naturaleza, son calculados con la misma longitud de caminos que conectan átomos y tienen valores similares. Dado que al incrementar el número de datos generalmente se producen

modelos QSPR más confiables se recomienda el uso del modelo de la Ec. (6.4.1) para la predicción de cualquier otro solvente en el futuro.

La estandarización de los coeficientes[74] de la regresión de la Ec. (6.4.1) permite asignar una importancia mayor a los descriptores que muestran un coeficiente estandarizado absoluto mayor (mostrado entre paréntesis):

$$Sp\ (2.669) > RDF015e\ (2.288) > Jhetv\ (0.875) > GATS5v\ (0.4369) \quad (6.4.4)$$

El ordenamiento dado por la Ec. (6.4.4) muestra que el descriptor electrónico *Sp* y la función de distribución radial *RDF015e* son las variables más relevantes para el presente grupo de solventes, indicando que el tiempo de luminiscencia (τ) del $Eu(fod)_3$ tendrá una dependencia significativa con la polarizabilidad y la electronegatividad del solvente.

6.4.4 Conclusiones

En este trabajo hemos construido un modelo QSPR del tiempo de vida de la luminiscencia (τ) del $Eu(fod)_3$ en 23 solventes diferentes con buen carácter predictivo para los cuales no se había llevado a cabo este tipo de análisis anteriormente, siendo τ una importante propiedad del $Eu(fod)_3$ el cual es usado en diversas y valiosas aplicaciones. Se usaron cuatro descriptores moleculares que tienen en cuenta aspectos bi y tridimensionales de la estructura molecular. Se presentó una nueva estrategia para la determinación del número optimo de variables que mostro funcionar exitosamente. Nuestros resultados sugieren que ERM y RM son preferibles a AG. El análisis del modelo encontrado sugiere que τ depende significativamente de la polarizabilidad y electronegatividad del solvente.

Los excelentes resultados obtenidos en el trabajo tuvieron como consecuencia su publicación en una importante revista: Mercader, A.G., P.R. Duchowicz, F.M. Fernández, E.A. Castro, E. Wolcan, Chemical Physics Letters, 462 (2008) 352–357. Es de esperar que este aporte a la comunidad científica ayude a dilucidar el mecanismo de desactivación del $Eu(fod)_3$ en distintos solventes lo que posiblemente lleve a encontrar nuevas aplicaciones.

Tabla 6.4.1 Valores experimentales y predichos (Ec. (6.4.1)) de tiempo de vida de luminiscencia τ del $Eu(fod)_3$ y los correspondientes residuos

Nº	Solvente	τ/ms exp.	τ/ms pred.	residuo
Conjunto de Calibración				
1	Ethanol	0.45	0.45	0.00
2	2-Propanol	0.43	0.48	-0.05
3	Bencylic alcohol	0.37	0.30	0.07
4	Dicholomethane	0.30	0.30	0.00
5	Chloroform	0.33	0.29	0.04
6	1,2-Dichloroethane	0.32	0.27	0.05
7	1,1-Dichloroethane	0.31	0.34	-0.03
8	Carbon tetrachloride	0.30	0.29	0.01
9	Toluene	0.25	0.29	-0.04
10	m-Xilene	0.34	0.31	0.03
11	p-Xilene	0.19	0.19	0.00
12	Benzene	0.35	0.33	0.02
13	Chlorobenzene	0.24	0.24	0.00
14	Bromobenzene	0.20	0.26	-0.06
15	Fluorobenzene	0.18	0.25	-0.07
16	Dioxane	0.56	0.56	0.00
17	Tetrahydrofurane	0.67	0.66	0.01
18	Ethyl ether	0.42	0.40	0.02
19	Acetone	0.58	0.59	-0.01
20	Cyclohexane	0.59	0.61	-0.02
21	Methyl ethyl ketone	0.54	0.54	0.00
22	Methyl propyl ketone (iso)	0.57	0.53	0.04
23	Acetonitrile	0.64	0.63	0.01
Conjunto de validación				
24	Methanol	0.45	0.40	0.05
25	n-Butanol	0.43	0.45	-0.02

Tabla 6.4.2 Valores incrementales de k y el resultante número de descripotres (d) que presenta un máximo en *VFIT*.

k	d (max. en *VFIT*)
1	-
1.5	-
2	9
2.5	7
3	4
3.5	4
4	4
4.5	3
5	2
5.5	2
6	1

Tabla 6.4.3 Modelos lineares QSPR para el conjunto de calibración con N=23. El mejor modelo está marcado en negrita

Modelo	**DESCRIPTORES**	***R***	***S***	***VFIT***	***RMSE$_{Val}$***
M1	*Mor16v*	0.831	0.0863	1.760	0.0665
M2	*Mor16v, nTB*	0.892	0.0716	2.317	0.0649
M3	*Sp, Jhetv, RDF015e*	0.942	0.0546	3.203	0.0425
M4	***GATS5v, Sp, Jhetv, RDF015e* (Ec.(6.4.1))**	**0.975**	**0.0375**	**4.856**	**0.0381**
M5	*Ss, Jhetp , X0Av, HOMA, RDF050m*	0.983	0.0319	4.090	0.1020
M4GA	*Mor30e, piPC05, BEHm5, X1Av*(Ec.(6.4.2))	0.968	0.0420	3.815	0.1246

Tabla 6.4.4 Símbolos de los descriptores usados en los diferentes modelos.

Descriptor	**Tipo**	**Definición**
Mor16v	3D-MoRSE	3D-MoRSE - signal 16 / weighted by atomic van der Waals volumes.
nTB	Constitutional	Number of triple bonds.
Sp	Constitutional	sum of atomic polarizabilities (scaled on Carbon atom).
Jhetv	Topological	Balaban-type index from van der Waals weighted distance matrix.

RDF015e	Radial Distribution Function	Radial Distribution Function - 1.5 / weighted by atomic Sanderson electronegativities.
GATS5v	2D Autocorrelations	Geary autocorrelation - lag 5 / weighted by atomic van der Waals volumes.
Ss	Constitutional	Sum of Kier-Hall electrotopological states.
Jhetp	Topological	Balaban-type index from polarizability weighted distance matrix.
X0Av	Topological	Average valence connectivity index chi-0.
HOMA	Aromaticity Indices	Harmonic Oscillator Model of Aromaticity index.
RDF050m	Radial Distribution Function	Radial Distribution Function - 5.0 / weighted by atomic masses.
Mor30e	3D-MoRSE	3D-MoRSE - signal 30 / weighted by atomic Sanderson electronegativities.
piPC05	Topological	Molecular multiple path count of order 05.
BEHm5	BCUT	Highest eigenvalue n. 5 of Burden matrix / weighted by atomic masses.
X1Av	Topological	Average valence connectivity index chi-1.

Tabla 6.4.5 Matriz de correlación para los descriptores de la Ec. (6.4.1) (N=23).

	GATS5v	Sp	Jhetv	RDF015e
GATS5v	1	0.6732	0.4017	0.2474
Sp		1	0.5003	0.7299
Jhetv			1	0.0862
RDF015e				1

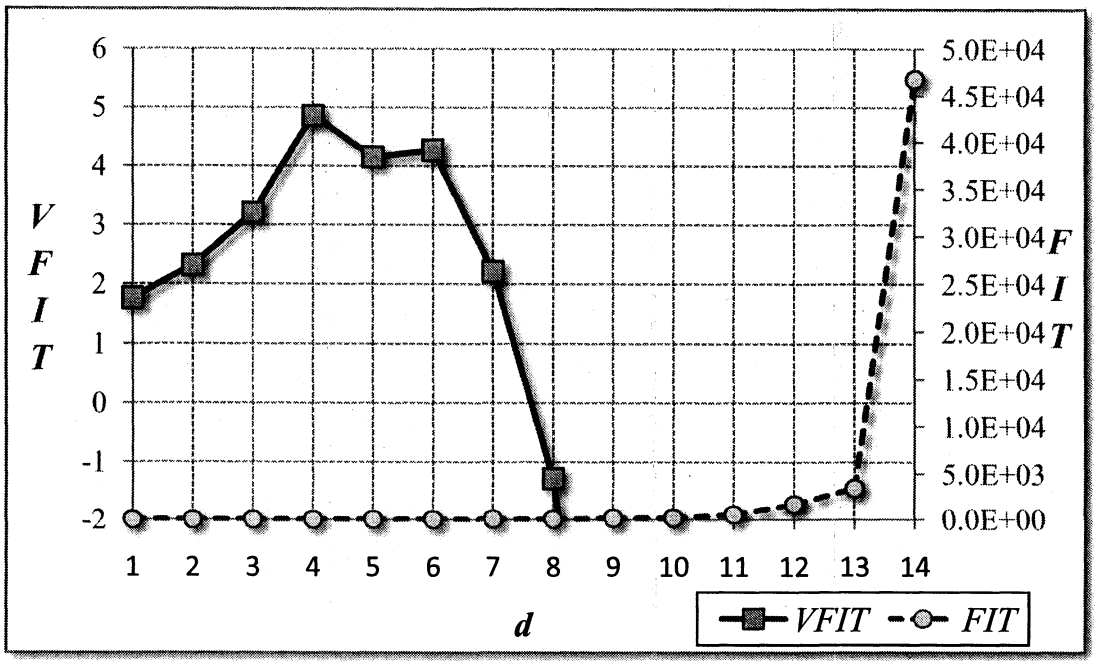

Figura 6.4.1 Parámetros *VFIT* (cuadrados) y *FIT* (círculos en el eje secundario) cómo función del número de descriptores para la calibración.

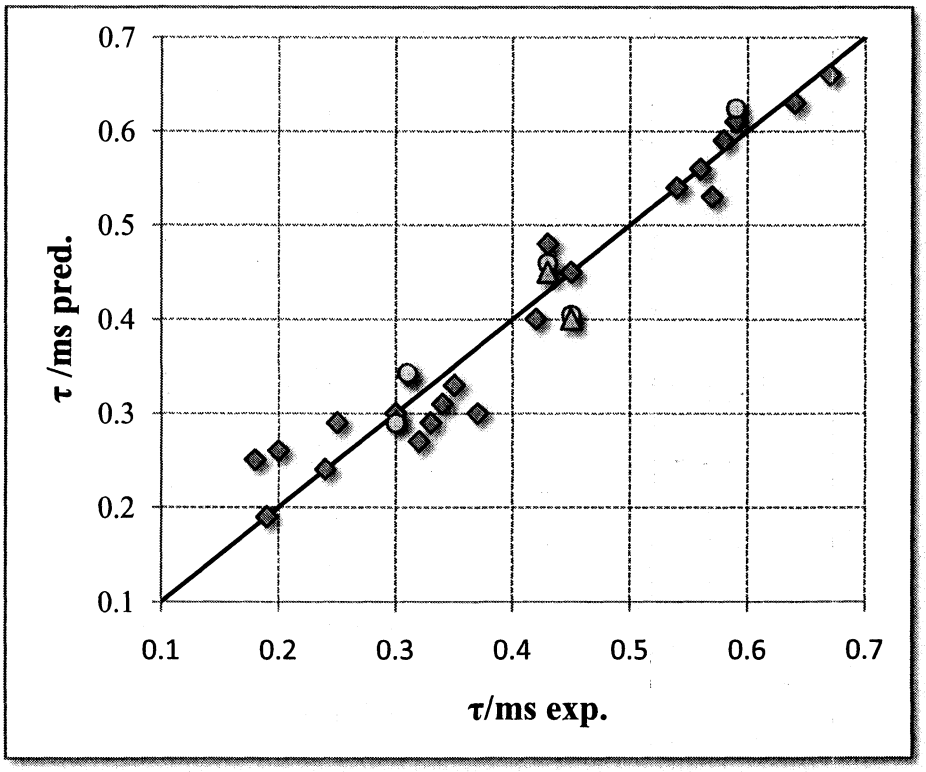

Figura 6.4.2 Tiempo de vida τ experimental versus predicho por Ec. (6.4.1) para el conjunto de calibración (rombos), conjunto de validación (triángulos), y predichos por Ec. (6.4.3) para el segundo conjunto de validación (círculos).

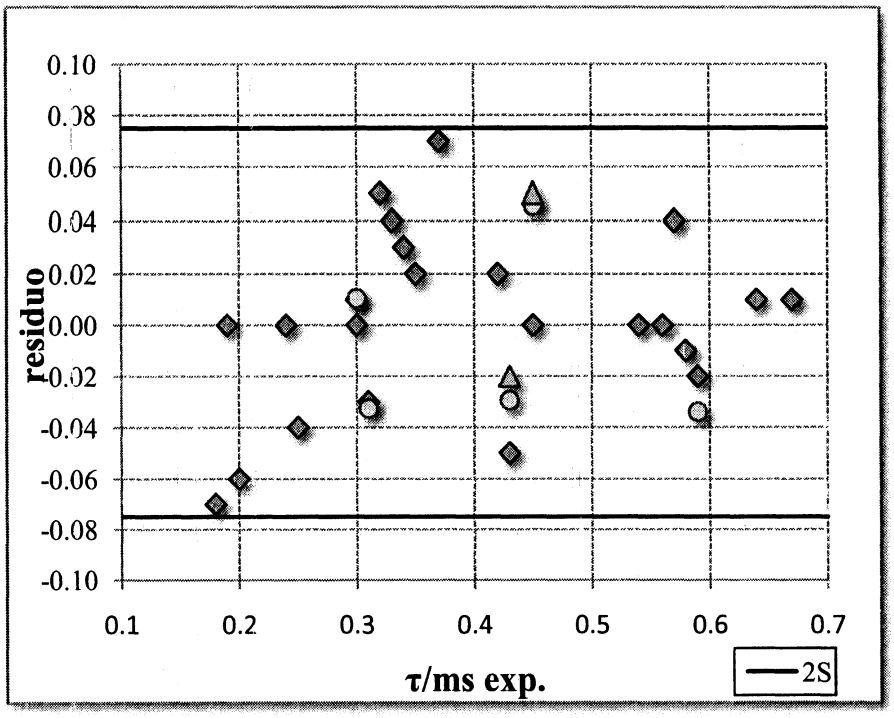

Figura 6.4.3 Gráfico de la dispersión de los residuos para los conjuntos de calibración y validación de acuerdo a la Ec. (6.4.1) y para el conjunto de validación usando la Ec. (6.4.3).

6.5 Estudio QSPR de Constantes de Desactivación Física y Química del Oxígeno Singlete por compuestos heterocíclicos

6.5.1 Introducción

El oxigeno está presente en el 50% de la corteza terrestre y es un componente esencial en las rutas metabólicas de todos los organismos complejos. [151] Con dos estados singletes por encima de su estado de menor energía triplete, la molécula de O_2 posee una configuración única, que da a lugar a importantes interacciones fotoquímicas.[152] El estado excitado más bajo del oxigeno molecular, llamado Oxigeno Singlete ($O_2(^1\Delta_g)$, indicado como $^1O_2$), es una molécula electrófila que tiene una alta capacidad de oxidar una gran variedad de compuestos orgánicos ricos en electrones.[153] Asimismo esta especie activa tiene propiedades físicas y químicas que han intrigado a investigadores científicos de numerosas áreas por más de 70 años;[154, 155] participando

en reacciones interesantes para diferentes campos de la ciencia como: química del medio ambiente, bromatología, bioquímica, biología, etc. A pesar de que el $^1O_2$ puede ser generado por vías químicas, fotoquímicas y por enzimas, la foto sensibilidad es el principal responsable de la producción de $^1O_2$ *in vivo*.[156] Mas aún, $^1O_2$ es una de las especies activas responsables de los efectos dañinos de la luz en sistemas biológicos.[155]

El $^1O_2$ se relaja a su estado fundamental $^3O_2$ por vías que emiten radiación y vías que no lo hacen:

$$^1O_2 \xrightarrow{k_d} {}^3O_2 \qquad (6.5.1)$$

$$^1O_2 \xrightarrow{k_e} {}^3O_2 + h\nu \qquad (6.5.2)$$

También pude ser desactivado por la oxidación de una molécula Q

$$Q + {}^1O_2 \xrightarrow{k_r} QO_2 \qquad (6.5.3)$$

y/o por su interacción física con la misma

$$Q + {}^1O_2 \xrightarrow{k_q} Q + {}^3O_2 \qquad (6.5.4)$$

Cualquier compuesto biológico que pueda desactivar eficientemente al $^1O_2$ pude tener un rol protector frente a $^1O_2$ *in vivo* y muy posiblemente frente a otras especies reactivas del oxígeno. Por lo tanto, el estudio de la reactividad del $^1O_2$ con bio-moléculas proveería su capacidad antioxidante. La determinación de la constante de desactivación total (física y química) del $^1O_2$ ($k_t = k_r + k_q$) permite la evaluación de la eficiencia de estos procesos.

La determinación experimental de los valores de k_t es un proceso complejo que requiere equipo especializado para detectar la débil señal de la luminiscencia del $^1O_2$ en el infrarrojo cercano.[157] Debido a esto, a pesar de la importancia bilogía del tema y la enorme diversidad de compuestos que pueden interactuar con el $^1O_2$, los estudios cinéticos de la reactividad de esta especie química todavía son escasos.

Los compuestos heterocíclicos se encuentran ampliamente distribuidos en los

sistemas biológicos y participan en funciones biológicas altamente relevantes. Familias importantes de bio-moléculas, como porfirinas, flavinas y pterinas pertenecen a este grupo de compuestos. En particular la reactividad del $^1O_2$ frente a pterinas fue recientemente estudiada.[158-160]

Claramente, es de gran interés la predicción de constantes de desactivación desconocidas del $^1O_2$ por un grupo de compuestos determinado, como así intentar determinar los parámetros estructurales de los que depende k_t. Una manera de llevar a cabo esto es mediante un análisis QSPR.[3]

En este trabajo se han predicho los valores de k_t para 41 compuestos heterocíclicos con diferentes grupos funcionales, cuyos datos experimentales fueron recolectados de la literatura.[160-162] Cabe mencionar que esta es la primera vez que dicho conjunto de moléculas es usado en un estudio QSPR.

Un gran número de descriptores moleculares que incluían definiciones de todas las clases se exploró usando el recientemente desarrollado ERM[82] Los resultados se compararon con los anteriormente aplicados RM[77-79, 116] y AG. [76]

Como una muestra práctica de las posibles aplicaciones de nuestro modelo QSPR se estimaron las k_t de compuestos heterocíclicos aún no medidos que resultaban particularmente interesantes para futuros estudios experimentales.

6.5.2 Métodos

6.5.2.1 Datos

El conjunto de calibración consistió en 41 compuestos heterocíclicos que tenían constate de desactivación del $^1O_2$ conocida.[160-162] De los datos experimentales disponibles se seleccionaron solo aquellos medidos en solventes polares próticos, a la misma temperatura (T= 298 K) y presión (P=1 atm). De acuerdo a los datos experimentales disponibles en la literatura[162] se asumió que diferentes solventes próticos no afectarían significativamente los constantes de desactivación. La Tabla 6.5.1 muestra los valores experimentales de $\log(k_t)$ para los compuestos heterocíclicos seleccionados.

6.5.2.2 Descriptores Moleculares

Como es habitual, las moléculas fueron pre optimizadas con un método de *Molecular Mechanics Force Field* (MM+), y luego se refinó la estructura resultante usando un método semi-empírico PM3 (*Parametric Method-3*) usando un algoritmo de Polak-Ribiere y un límite de gradiente de 0.01 kcal.Å-1. En este caso se usó el software e-Dragon,[163] para calcular los descriptores moleculares a los cuales se les adicionaron 20 descriptores constitucionales que tiene en cuenta grupos funcionales y su posición en la molécula; y cuatro derivados de la química-cuántica (momento dipolar molecular, energías totales y energías de HOMO-LUMO) no incluidos en el software antes mencionado, resultando un grupo de D=1659 descriptores moleculares de distintas clases.[120]

Se usaron validaciones cruzadas *l-o-o* y *l-n%-o*[36], con *n%=30%* (12 compuestos heterocíclicos) siendo este el número de moléculas removidas del grupo de calibración, 5000000 fue el número de casos aleatorios empleados.

6.5.3 Resultados y Discusión

Por medio de ERM llevamos a cabo una búsqueda entre el conjunto total de descriptores D=1659 y se obtuvieron modelos óptimos con d=1,2,...,15 parámetros que vinculan la estructura molecular de los compuestos heterocíclicos con la constante de desactivación total k_t.

Para determinar el número óptimo de parámetros se empleó el novedoso método que hace uso del parámetro *VFIT* presentado en la sección 4.7. En este caso a medida que k en *VFIT* es incrementado (ver Tabla 6.5.2) se encuentran cuatro máximos distintos en d=13 (k=2), d=9 (k=3), d=8 (k=3.5) y d=6 (k=4). Este último cumple con la regla que especifica que el numero de parámetros para este estudio debería ser menor a 8.[98] Por lo tanto este sería el número optimo de descriptores para el modelo. La Figura 6.5.1 muestra el máximo en *VFIT* con k=4 ($d=d_{max}$=6) y también que FIT no presenta un máximo en el intervalo $1 \leq d \leq 15$. En la Tabla 6.5.2 se puede apreciar que d_{max}=6 permanece constante frente a incrementos adicionales de k aseverando el hecho de que este es el número optimo de descriptores.

Por lo tanto, al usar ERM el modelo óptimo QSPR encontrado fue:

$$\log(k_l) = 7.8611(\pm 0.4) - 1.2329(\pm 0.2)\cdot GATS1e - 1.1219(\pm 0.1)\cdot Mor16u + 5.5727(\pm 0.9)\cdot E1v - 32.9421(\pm 3.8)\cdot R8m^{+} - 1.4465(\pm 0.1)\cdot nN^{+} - 1.1799(\pm 0.1)\cdot nArOH \quad (6.5.5)$$

$$N = 41,\ R = 0.9727,\ S = 0.2323,\ FIT = 7.7689,\ p < 10^{-5}$$
$$R_{loo} = 0.9609,\ S_{loo} = 0.2778,\ R_{l-30\%-o} = 0.8793,\ S_{l-30\%-o} = 0.4892$$

donde, los errores absolutos de la regresión se encuentran en paréntesis y p es la significancia del modelo.

Haciendo la búsqueda con RM[77-79, 116] para un grupo de d=6 descriptores se llegó al siguiente modelo:

$$\log(k_l) = 6.8842(\pm 0.9) + 0.7754(\pm 0.1)\cdot nR10 - 0.803(\pm 0.2)\cdot GATS1p + 42687(\pm 1.4)\cdot E1e - 23.779(\pm 4.3)\cdot R8m^{+} - 1.5004(\pm 0.2)\cdot nN^{+} - 0.4449(\pm 0.03)\cdot N - 075 \quad (6.5.6)$$

$$N = 41,\ R = 0.9603,\ S = 0.2797,\ FIT = 5.2249,\ p < 10^{-3}$$
$$R_{loo} = 0.9385,\ S_{loo} = 0.3474,\ R_{l-30\%-o} = 0.7993,\ S_{l-30\%-o} = 0.6075$$

Como punto de comparación se usó un AG para seleccionar un modelo de d_{opt} = 6 descriptores. Para poder hacer esto se optimizaron los parámetros del AG para este problema en particular llevando a cabo numerosas pruebas encontrando los siguientes ajustes: Número de individuos = 250; Brecha generacional = 0.9; Probabilidad de entrecruzamiento = 0.6; Probabilidad de mutación = 0.7/d. El criterio para frenar la evolución fue determinado de forma tal que cuando un individuo ocupara más del 90% de la población o cuando el número de generaciones fuese 2500 se interrumpiría el proceso. El mejor modelo encontrado usando AG fue:

$$\log(k_l) = 7.1903(\pm 0.5) - 1.533(\pm 0.1)\cdot C\text{-}032 - 23.733(\pm 3.9)\cdot R8m^{+} - 1.3538(\pm 0.2)\cdot nN^{+} + 5.4508(\pm 1.2)\cdot De - 1.1976(\pm 0.2)\cdot GATS1p + 0.8661(\pm 0.1)\cdot nR10 \quad (6.5.7)$$

$$N = 41,\ R = 0.968,\ S = 0.2515,\ FIT = 6.5671,\ p < 10^{-4}$$
$$R_{loo} = 0.9524,\ S_{loo} = 0.306,\ R_{l-30\%-o} = 0.4464,\ S_{l-30\%-o} = 7.6249$$

Estos resultados muestran que ERM es superior a AG y RM para una búsqueda en un gran conjunto de descriptores. La Tabla 6.5.3 muestra un resumen de los modelos lineales con un número de parámetros que va de 1 a d_{opt} +1 para ERM y d_{opt} parámetros para RM y GA. En esta tabla se puede apreciar que mientras que los parámetros de calibración continúan mejorando, los parámetros estadísticos de validación mejoran al aumentar d hasta llevgar a d_{opt} para luego deteriorarse, esto corrobora la bondad del método usado para seleccionar d_{opt}. Los detalles de los descriptores moleculares de la Tabla 6.5.3 fueron presentados en la Tabla 6.5.4.

La matriz de correlación de la Tabla 6.5.5 indica que los descriptores del modelo lineal no están importantemente inter correlacionados (R_{ij}<0.5334), y esto sustenta la presencia de todos los parámetros en la ecuación. El poder predictivo del modelo es satisfactorio frente a la inclusión y exclusión de compuestos, esto fue medido con los parámetros estadístico de R_{loo} = 0.9609 y $R_{l\text{-}30\%\text{-}o}$ = 0.8793. De acuerdo a la literatura $R_{l\text{-}n\%\text{-}o}$ debe ser mayor que 0.71 para poder considerar un modelo como correctamente validado.[104]

Se utilizó el método de la variable Y aleatoria[105] para demostrar que los resultados obtenidos en Ec. (6.5.5) no eran fortuitos (para más detalles sobre el método ver sección 5.3.2). Luego de analizar 5000000 de casos el menor valor obtenido es S = 0.5691 siendo este considerablemente mayor al obtenido de la calibración S = 0.2323. Esto sugiere que el modelo encontrado es robusto y que la relación estructura actividad es confiable. Este resultado junto con el resto de los parámetros estadísticos de calibración y validación están de acuerdo con la premisa que los diferentes solventes polares próticos no afectarían significantemente las medidas de k_t.

El gráfico de valores predichos versus experimentales de log(k_t) mostrado en la Figura 6.5.2 indica que los 41 compuestos heterocíclicos siguen aproximadamente una línea recta. La Tabla 6.5.1 también incluye los valores de log(k_t) predichos por la Ec. (6.5.5) y los residuos correspondientes. La Figura 6.5.3 muestra que el comportamiento de los residuos tiene una distribución normal, asimismo no hay moléculas que presenten un residuo mayor a 2.5S que puedan ser consideradas como *outliers*.

Se llevó a acabo la prueba adicional del poder predictivo del modelo, presentada anteriormente en la sección 5.3.3. En este caso se quitaron 6 moléculas (con valores variados de la propiedad) del conjunto de calibración y se recalculó el modelo obteniéndose la siguiente ecuación basada en las 35 moléculas remanentes:

$$\log(k_t) = 7.9238(\pm 0.5) - 1.3282(\pm 0.2)\cdot GATS1e - 1.1456(\pm 0.1)\cdot Mor16u + 5.7009(\pm 1)\cdot E1v - 32.2832(\pm 4.3)\cdot R8m^{+} - 1.4369(\pm 0.1)\cdot nN^{+} - 1.1991(\pm 0.1)\cdot nArOH \quad (6.5.8)$$

$$N = 35,\ R = 0.9749,\ S = 0.2352,\ FIT = 7.5683,\ p < 10^{-5}\ RMSE_{Val} = 0.2325$$

Los parámetros son similares a los de la Ec. (6.5.5), $RMSE_{Val}$ significa raíz del error cuadrático medio del conjunto de validación compuesto por las 6 moléculas antes mencionadas. Las Figuras 6.5.2 y 6.5.3 muestran que el gráfico de valores $\log(k_t)$ experimentales frente a los predichos para los 6 compuestos heterocíclicos siguen aproximadamente una línea recta y no hay moléculas que puedan considerarse como *outliers*.

Con el propósito principal de estimar si el conjunto de validación seleccionado puede ser usado como un conjunto realmente externo para la Ec. (6.5.8), se llevó a cabo una nueva búsqueda usando ERM para el nuevo conjunto de calibración de 35 moléculas antes definido. El mejor modelo encontrado es casi idéntico al de la Ec. (6.5.5) (basada en el conjunto completo de 41 moléculas) y posee parámetros estadísticos similares de validación y calibración. La única diferencia es que el descriptor *GATS1v* reemplazó a *GATS1e*; estos dos descriptores tienen la misma naturaleza, son calculados con la misma distancia entre átomos y tienen valores similares. En consecuencia, ya que un incremento en el número de datos experimentales permite obtener modelos QSPR más confiables, se recomienda usar la Ec. (6.5.5) para cualquier predicción futura de la propiedad en estudio ya que esta codifica información de todo el conjunto de 41 moléculas.

La estandarización de los coeficientes de regresión[74] de la Ec. (6.5.5) permite asignar una importancia mayor a los descriptores que muestran un coeficiente estandarizado absoluto mayor (mostrado entre paréntesis):

$$nArOH(0.5353) > Mor16u(0.5036) > nN^{+}(0.4704) > R8m^{+}(0.4151) > E1v(0.3352) > GATS1e(0.2959) \quad (6.5.9)$$

El ordenamiento dado por la Ec. (6.5.9) muestra que los descriptores de cuenta de grupos funcionales *nArOH* y nN^{+} son las variables más importantes. El descriptor tridimensional más relevante es del tipo 3D-MoRSE y se llama *Mor16u*; este tipo de descriptores normalmente codifica información tridimensional en una forma eficiente lo que implicaría una dependencia significativa frente a cambios conformacionales[66, 67].

Por medio de la Ec. (6.5.5) se estimaron las constantes k_t para un grupo de heterociclos llamados β-carbolinas (Figura 6.5.4). Más allá del hecho de que han sido sugeridas como posibles antioxidantes,[164] hasta el momento no existen datos experimentales de k_t para estas. En soluciones acuosas, las β-carbolinas presentan un equilibrio acido-base con un pKa de aproximadamente 7. Dado que las β-carbolinas están presentes en un gran numero de sistemas vivos, con un pH fisiológico que ronda 7, se decidió calcular las constantes para las formas ácida y básica. Los resultados se encuentran expuestos en la Tabla 6.5.6.

6.5.4 Conclusiones

Por medio del algoritmo de búsqueda ERM se construyó un modelo QSPR para la predicción de la constante de desactivación del oxigeno singlete usando datos de 41 compuestos heterocíclicos. El modelo encontrado constó de seis descriptores moleculares que tienen en cuenta aspectos bi y tri dimensionales de la estructura molecular. Los resultados indican que ERM es superior a RM y los AG. La estrategia novedosa para determinar el número óptimo de descriptores fue exitosamente usada. Haciendo uso del modelo QSPR se estimaron las constantes para β-carbolinas para las cuales no existían valores experimentales hasta el momento.

Los excelentes resultados obtenidos en el trabajo serán publicados a la brevedad: Mercader, A.G., P.R. Duchowicz, F.M. Fernández, E.A. Castro, F.M. Cabrerizo, A.H. Thomas, Predictive Modeling of the Total Deactivation Rate Constant of Singlet Oxygen by Heterocyclic Compounds. Journal of Molecular Graphics and Modelling, 2009.(Presentado)

Es de esperar que este aporte a la comunidad científica ayude a encontrar nuevos compuestos con propiedades antioxidantes.

Tabla 6.5.1 Datos experimentales y predichos de $\log(k_i)$ por la Ec. (6.5.5) y los residuos.

Número	Nombre	$\log(k_i)$ exp.	$\log(k_i)$ pred.	residuo
1	7,8-dihydrofolic acid	8.74	8.59	0.15
2	7,8-dihydrobiopterin	8.57	8.28	0.29
3	7,8-dihydroneopterin	8.66	8.71	-0.05
4	6-formyl-7,8-dihydropterin	8.32	8.19	0.13
5	sepiapterin	8.28	8.28	0.00
6	7,8-dihydroxantopterin	8.83	9.09	-0.26
7	pterin	6.46	6.92	-0.46
8	6-methylpterin	6.90	6.91	-0.01
9	6,7-dimethylpterin	7.60	7.25	0.35
10	6-(hydroxymethyl)pterin	6.49	6.21	0.28
11	6-formylpterin	6.15	6.23	-0.08
12	6-carboxypterin	6.15	6.22	-0.07
13	biopterin	6.38	6.35	0.03
14	neopterin	6.36	6.36	0.00
15	folic acid	7.48	7.52	-0.04
16	histamina	8.06	8.21	-0.15
17	imidazole	7.46	7.31	0.15
18	4-methyl-imidazole	8.11	7.89	0.22
19	indole	7.65	8.00	-0.35
20	2,3-dimethy-indole	8.76	8.65	0.11
21	3-methyl-indol	8.20	8.33	-0.13
22	indole 3 acetic acid	8.83	8.49	0.34
23	indole-3-propianamide	7.89	8.07	-0.18
24	indole-3-propionic acid	7.91	8.31	-0.40
25	2,5-diphenyl-oxazole	8.20	8.05	0.15
26	2,5-diphenyl-4-methyl-oxazole	7.53	7.68	-0.15

27	4-methyl-2-(3-chlorophenyl)-5-phenyl oxazole	7.23	7.23	0.00
28	4-methyl-2-(4-chlorophenyl)-5-phenyl oxazole	7.28	7.43	-0.15
29	4-methyl-2-(4-methoxyphenyl)-5-phenyl oxazole	7.72	7.59	0.13
30	4-methyl-2-(4-methylphenyl)-5-phenyl oxazole	7.57	7.67	-0.10
31	4-methyl-2-(4-nitrophenyl)-5-phenyl oxazole	7.08	7.01	0.07
32	2,3-dihydro-1-methyl-4-phenyl-pyridinium	6.23	6.09	0.15
33	1-methyl-pyridinium	5.81	6.14	-0.33
34	1-methyl-4-phenyl-pyridinium	5.95	5.83	0.12
35	cis(-)-2,3,4,4a,5,9b-hexahydro-2,8-dimethyl-pyrido[4,3-b]indole	8.11	8.37	-0.26
36	1,2,3,4-tetrahydro-2,8-dimethyl-pyrido[4,3-b]indole	8.23	8.46	-0.23
37	1-(1,1-dimethylethyl)-pyrrole	8.08	7.89	0.19
38	2-(1,1-dimethylethyl)-pyrrole	8.18	8.28	-0.10
39	3-(1,1-dimethylethyl)-pyrrole	8.26	8.21	0.05
40	quinoline	9.00	8.51	0.49
41	1,2-dihydro-2,2,4-trimethyl-quinoline, homopolymer	8.98	8.86	0.12

Tabla 6.5.2 Valores de *k* y *d* correspondientes al máximo en *VFIT*

k	***d* (max.)**	***k***	***d* (max.)**	***k***	***d* (max.)**
1	-	6.5	3	12	2
1.5	-	7	3	12.5	2
2	13	7.5	2	13	2
2.5	13	8	2	13.5	2
3	9	8.5	2	14	2
3.5	8	9	2	14.5	2
4	6	9.5	2	15	2
4.5	6	10	2	15.5	2
5	4	10.5	2	16	2
5.5	4	11	2	16.5	2
6	4	11.5	2	17	1

Tabla 6.5.3 Modelos QSPR calculados para el conjunto de validación completo N=41

Modelo	DESCRIPTORES USADOS	R	S	$R_{l\text{-}30\%\text{-}o}$	$S_{l\text{-}30\%\text{-}o}$
M1	*C-027*	0.664	0.699	0.195	1.099
M2	*nN^+, N-075*	0.864	0.477	0.628	0.762
M3	*nR05, nR10, N-075*	0.906	0.406	0.350	1.299
M4	*nR10, SRW05, MATS5p, N-075*	0.939	0.334	0.227	3.258
M5	*GATS8m, GATS1p, Mor16v, nN^+, nArOH*	0.953	0.300	0.789	0.610
M6	***GATS1e, Mor16u, E1v, $R8m^+$, nN^+, nArOH* (Ec. (6.5.5))**	**0.973**	**0.232**	**0.879**	**0.489**
M7	*nR05, nR10, ATS8m, Mor29e, E1v, H5u, nArNH2*	0.980	0.204	0.509	2.709
M6B	*nR10, GATS1p, E1e, $R8m^+$, nN^+, N-075* (Ec. (6.5.6))	0.960	0.280	0.799	0.608
M6C	*C-032, $R8m^+$, nN^+, De, GATS1p, nR10* (Ec.(6.5.7))	0.968	0.251	0.446	7.625

Tabla 6.5.4 Definiciones de los descriptores que aparecen en los distintos modelos

Descriptor	Tipo	Definición
C-027	Atom-Centred Fragments	*C-027* corresponds to: R--CH--X
nN^+	Functional Group Counts	Number of ammonium groups (aliphatic)
N-075	Atom-Centred Fragments	*N-075* corresponds to: R--N--R / R--N--X.
nR05	Constitutional Descriptors	Number of 5-membered rings.
nR10	Constitutional Descriptors	Number of 10-membered rings.
SRW05	Molecular Walk Counts	Self-returning walk count of
MATS5p	2D Autocorrelations	Moran autocorrelation - lag 5 / weighted by atomic polarizabilities.
GATS8m	2D Autocorrelations	Geary autocorrelation - lag 8 / weighted by atomic masses.
GATS1p	2D Autocorrelations	Geary autocorrelation - lag 1 / weighted by atomic polarizabilities.
Mor16v	3D-MoRSE	3D-MoRSE - signal 16 / weighted by atomic van der Waals volumes.
nArOH	Functional Group Counts	Number of aromatic hydroxyls.
GATS1e	2D Autocorrelations	Geary autocorrelation - lag 1 / weighted by atomic Sanderson

		electronegativities.
Mor16u	3D-MoRSE	3D-MoRSE - signal 16 / unweighted.
E1v	WHIM	1st component accessibility directional WHIM index / weighted by atomic van der Waals volumes.
R8m^{+}	GETAWAY	R maximal autocorrelation of lag 8 / weighted by atomic masses.
ATS8m	2D Autocorrelations	Broto-Moreau autocorrelation of a topological structure - lag 8 / weighted by atomic masses.
Mor29e	3D-MoRSE	3D-MoRSE - signal 29 / weighted by atomic Sanderson electronegativities.
H5u	GETAWAY	H autocorrelation of lag 5 / unweighted.
nArNH2	Functional Group Counts	Number of primary amines (aromatic)
E1e	WHIM	1st component accessibility directional WHIM index / weighted by atomic Sanderson electronegativities.
C-032	Atom-Centred Fragments	*C-032* corresponds to: X--CX--X
De	WHIM	D total accessibility index / weighted by atomic Sanderson electronegativities.

Nota: para los descriptores Atom-Centred Fragments: R representa cualquier grupo unido por un carbón; X cualquier átomo electronegativo (O, N, S, P, Se, halógenos); -- representa un enlace aromático como en el benceno o un enlace de localizado como el N-O de un grupo nitro.

Tabla 6.5.5 Matriz de correlación para los descriptores de la Ec. (6.5.5) (N=41).

	GATS1e	Mor16u	E1v	R8m+	nN+	nArOH
GATS1e	1	0.1651	0.3444	0.3685	0.3021	0.0160
Mor16u		1	0.5334	0.0566	0.2129	0.2327
E1v			1	0.2439	0.0863	0.3577
R8m+				1	0.1114	0.2957
nN+					1	0.1744
nArOH						1

Tabla 6.5.6 Valores predichos de $\log(k_t)$ por la Ec. (6.5.5) para el grupo de moléculas sin datos experimentales.

Número	Nombre	$\log(k_t)$ pred.	k_t (L mol-1 s-1).
42	norharmane	8.69	4.90E+08
43	norharmane$^+$	7.01	1.03E+07
44	harmane	8.65	4.50E+08
45	harmane$^+$	7.12	1.32E+07
46	harmine	8.30	1.99E+08
47	harmine$^+$	6.69	4.89E+06
48	harmaline	8.17	1.47E+08
49	harmaline$^+$	6.46	2.90E+06
50	harmol	7.65	4.42E+07
51	harmol$^+$	6.16	1.43E+06

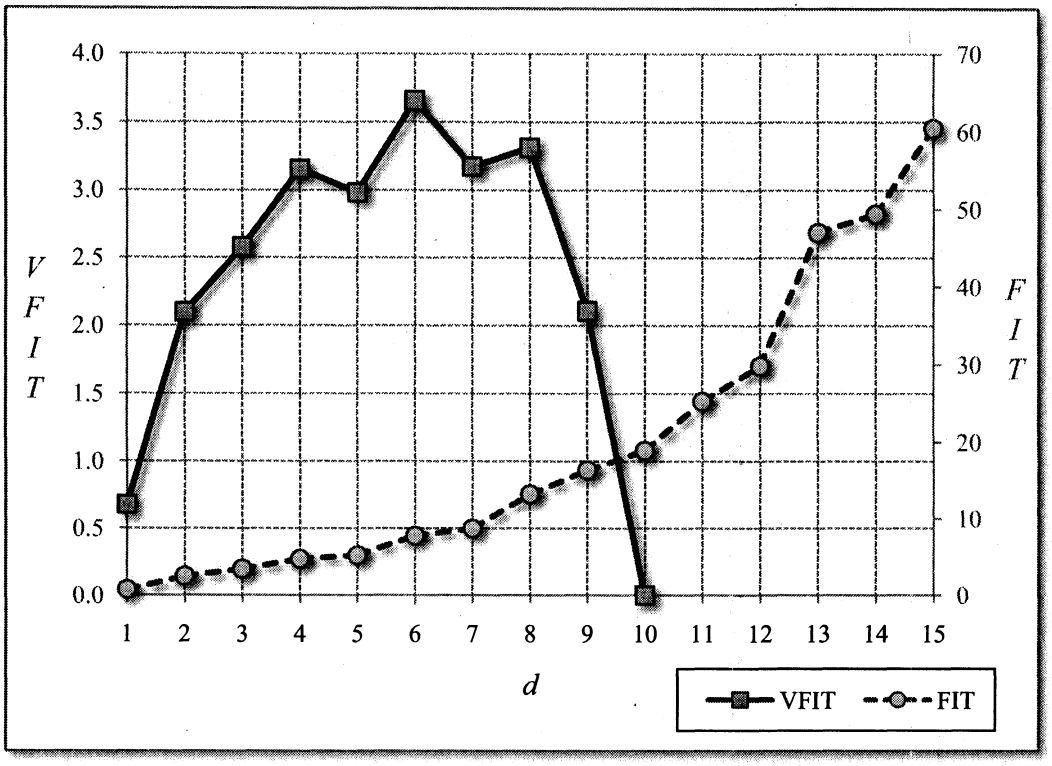

Figura 6.5.1 *VFIT* (cuadrados eje izquierdo) y *FIT* (círculos, eje derecho) en términos del número de descriptores para modelar el conjunto de calibración

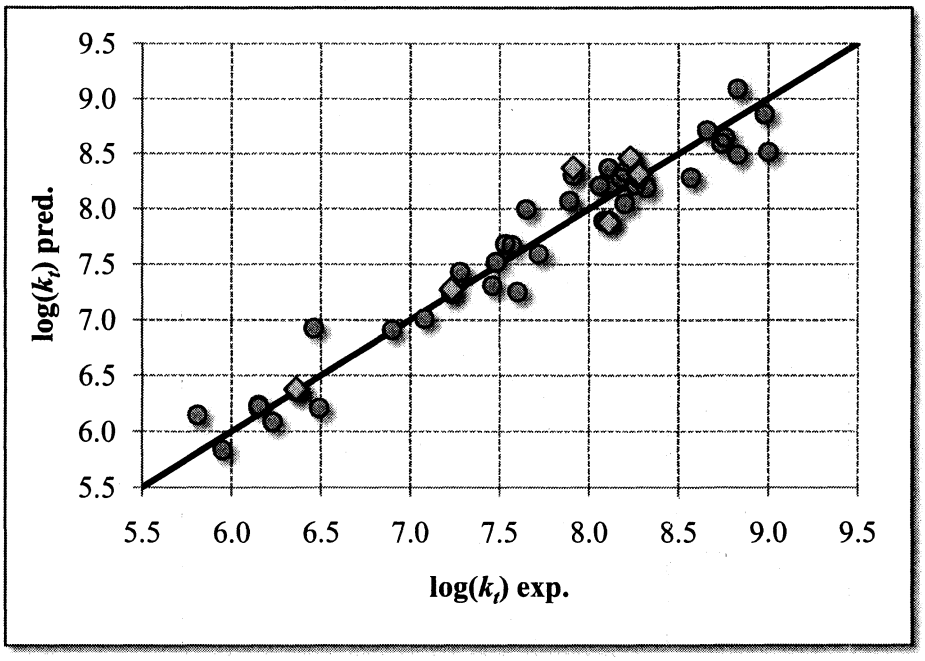

Figura 6.5.2 Datos experimentales de log(k_t) versus los predichos. Resultados de la Ec. (6.5.5) (círculos) y de la Ec. (6.5.8) para el conjunto de validación (rombos).

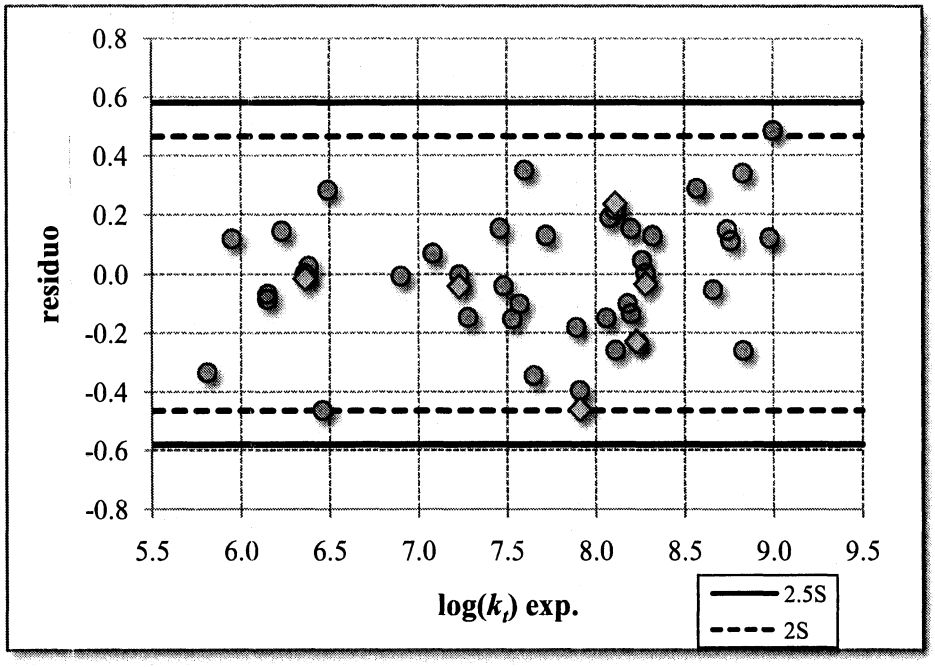

Figura 6.5.3 Gráfico de la dispersión de los residuos para los conjuntos de calibración a la Ec. (6.5.5) (círculos) y para el conjunto de validación usando la Ec. (6.5.8) (rombos).

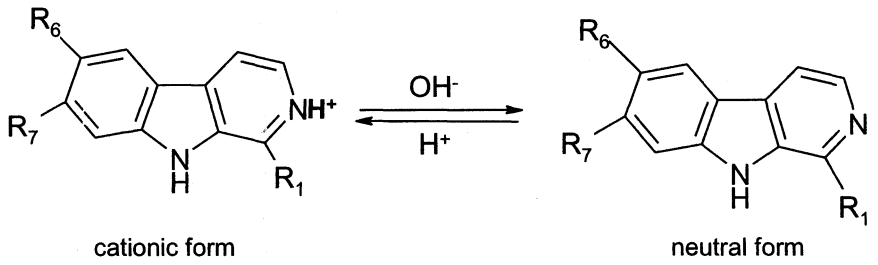

Figura 6.5.4 Estructura del equilibrio acido base en solución acuosa de las β-carbolinas

7 Mejoras computacionales

"La ciencia será siempre una búsqueda, jamás un descubrimiento real. Es un viaje, nunca una llegada" (Karl R. Popper)

7.1 Introducción

En las teorías QSAR/QSPR, como cualquier otra teoría que utilice recursos computacionales para realizar cálculos, cuando se exige cada vez más llega a un punto que uno se encuentra con algún tipo de limite[1]. Esto hace que el cálculo sea imposible de realizar, ya sea porque el software usado no lo puede manejar y produce un error, o porque los tiempos de cálculo se hicieron demasiado extensos. Para sobre llevar esto se utilizan métodos aproximados que reducen notablemente el uso de los recursos computacionales; un ejemplo podría ser lo visto en la sección 4 para el caso de métodos de búsqueda de descriptores. La cantidad de regresiones que se deben realizar usando métodos aproximados es notablemente menor a la búsqueda exhaustiva, claro está que se debe tratar de no reducir apreciablemente la calidad de los modelos encontrados.

Otro camino a seguir para poder realizar cálculos que para finalizar tomen una cantidad de tiempo razonable es la optimización del uso del procesador, es decir reducir el tiempo de cálculo sin necesidad de reducir el número de cálculos. Una ventaja de seguir este camino es que además la mejora se ve reflejada en el resto de los cálculos necesarios para estudios QSAR/QSPR[165], como la validación cruzada *lmo* y la elección aleatoria de *y* (ver sección 5); estas últimas necesitan un número muy grande de cálculos para tener validez estadística.

7.2 Optimización del Software utilizado

La primera pregunta que uno se hace a la hora de comenzar a programar es qué software o idioma informático es mejor usar.

Para responder eso se deben tener en cuenta distintos aspectos.

Al iniciarse el trabajo de tesis los algoritmos disponibles en el grupo estaban programados en Derive[166], ya que es un sistema de álgebra computacional que está diseñado para realizar cálculos analíticos y trabajar con números racionales e irracionales de manera exacta. El mismo permite invertir matrices en forma exacta lo cual permite superar el problema de matrices casi singulares sin tener que escribir algoritmos especiales para ello. Sin embargo, los requerimientos computacionales de los estudios QSAR/QSPR fueron incrementandose notablemente a lo largo de los últimos años haciendo que los tiempos de cálculo llegaran a ser muy importantes; ocasionando que muchas veces fuera necesario esperar varios días para que finalice una corrida. Era sabido que si se usaba alguno de los lenguajes de programación basado en operaciones de punto flotante el tiempo de cálculo se reduciría apreciablemente lo que motivó la traducción de los algoritmos a MATLAB.

La idea inicial era reescribir los algoritmos en un nivel de programación más bajo, ya que normalmente de esta forma se optimizan los recursos computacionales. Por eso se asistió a un curso de programación en el idioma ANSI C[167].

Sin embargo luego de finalizado el mismo se vió que la mejor opción teniendo en cuenta otros aspectos era programar en MATLAB[128]. Cabe mencionar que la estructura del lenguaje de MATLAB está basada en ANSI C por lo que los conocimientos adquiridos sobre este último fueron indispensables.

La decisión de usar MATLAB se tomó luego de obtener la experiencia de programadores alrededor del mundo usando información proveniente de diversos foros encontrados a través de internet. La conclusión a la qué se llegó es que para poder optimizar un programa en ANSI C era necesario no solo tener conocimientos básicos sino varios años de experiencia en programación en este lenguaje, cosa de la que se carecía. En cambio, para el tipo de algoritmos necesarios, donde básicamente se deben manejar matrices (ver sección 4), MATLAB haría la programación mucho más directa. El ahorro de tiempo estimado según los especialistas iba a ser aproximadamente de 10 a 1 programando en MATLAB versus programación en ANSI C. Adicionalmente si bien los algoritmos obtenidos no estarían optimizados para la aplicación particular, el sistema MATLAB se encuentra muy desarrollado y optimizado en el manejo de matrices por lo que el uso de los recursos computacionales en este aspecto era óptimo. Cabe resaltar que además el manejo de resultados e ingreso de datos es mucho más simple en MATLAB que en ANSI C.

7.3 Resultados logrados

Para adquirir los conocimientos básicos en el sistema MATLAB y poder comenzar a incursionar en el mismo, se utilizó un tutorial disponible en la web[168].

La traducción de algoritmos fue un proceso un tanto laborioso al comienzo y se fue acelerando al ir ganando experiencia en programación. El resultado final fue exitoso, las pruebas demostraron que tal cómo era esperado se obtenían exactamente los mismos resultados que usando los algoritmos del Derive. El avance más notable se vió en el tiempo de cálculo necesario, el mismo se vió reducido increíblemente superando ampliamente las expectativas. La reducción del tiempo de cálculo depende del cálculo realizado, a mayor exigencia en el cálculo mayor es el ahorro del tiempo. Como se puede ver en la Tabla 7.3.1 el tiempo utilizado por MATLAB respecto al Derive para estos problemas a resolver es de 140 a 700 veces menor. Si se aumentase la exigencia computacional el ahorro de tiempo relativo sería aún mayor, ya que el MATLAB es aún mejor para cálculos más exigentes. Todos los cálculos se realizaron en la misma PC con un procesador AMD Athlon 64 2800+, memoria DDR de 1GB.

Para intentar explicar el aumento del ahorro de tiempo relativo se buscó el mejor ajuste de los datos, se pude observar que el aumento del tiempo con el número de regresiones tiene un comportamiento polinomial en el caso de MATLAB y potencial en el caso del Derive, ver Figura 7.3.1. Cabe mencionar que en el caso del Derive para lograr el mejor ajuste fue necesario restar una constante igual a 39.87 segundos a todos los datos de la Tabla 7.3.1, se estima que este tiempo es el que necesita el programa para ponerse en marcha el cual es independiente del número de regresiones. Este tiempo de arranque constante del Derive es lo que hace que haya una mejora en el valor relativo respecto al MATLAB entre los casos d=1 *y* d=2.

Tabla 7.3.1 Comparación tiempo de cálculo Derive vs MATLAB para distintos *d*. D=1269, N=116

d	N° de regre.	T Derive (s)	T MATLAB (s)	Derive/MATLAB
1	1269	41	0.125	330.4
2	20290	270	2.0	137.6
3	53175	1541	5.5	280.2
4	80964	3850	9.8	394.8
5	126405	6382	16.8	378.9
6	189456	15170	28.1	540.3
7	247359	26647	37.5	710.6
8	322824	38345	54.8	699.7

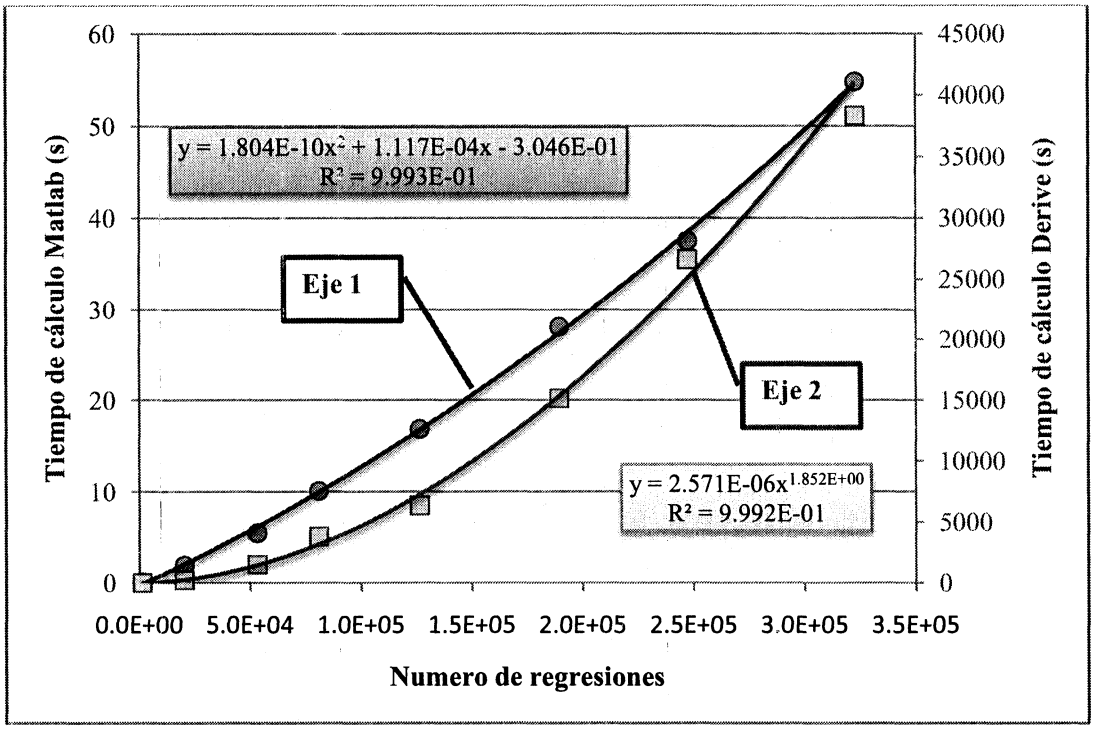

Figura 7.3.1 Comparación de tiempos de cálculo de MATLAB vs Derive

La enorme disminución del tiempo de cálculo luego de la optimización de los algoritmos tuvo importantes beneficios:

- Importante aceleración de los trabajos de investigación
- Mejoras en la calidad de los mismos:

- Permitió realizar pruebas que antes eran imposibles lo que llevó a obtener un algoritmo mejorado ERM[82] que si bien es computacionalmente un poco más exigente logró resultados aún mejores que RM[77]
- Se pudo aumentar notablemente el número de casos en las validaciones cruzadas (de 100.000 a 5.000.000)
- Se pudo realizar la elección aleatoria de **y** para 5.000.000 de casos
- Se pudieron elegir modelos de mayor número de descriptores (d=1 a d=14), lo que ayudó a mejorar el método de elección de número de variables optimas (ver sección 4.6.4)

- Además el sistema MATLAB trajo otras ventajas como:
  - Facilidad del ingreso y manejo de matrices provenientes del software de cálculo de descriptores Dragon[119]
  - Posibilidad de analizar y visualizar la evolución de los algoritmos paso a paso con mayor facilidad
  - Disminución del tiempo de apertura de archivos en la PC

7.4 Algoritmos Disponibles

Todo el código de utilidad (parte de las pruebas fueron descartadas) que fue programado durante el trabajo de tesis se deja disponible en el Apéndice (Sección 8), el objetivo de esto es facilitar el trabajo de cualquier persona interesada en realizar estudios QSAR/QSPR.

Los programas en MATLAB son escritos en un archivo de texto con extensión ".m" por lo tanto si se copian del Apéndice, se nombra el archivo con el mismo nombre que la función y se le cambia la extensión estos funcionarán perfectamente en el sistema MATLAB, las instrucciones detalladas se encuentran en la sección 8.2.

El idioma de los comentarios de los archivos de MATLAB se mantuvo en Ingles, ya que el mismo fue necesario para la publicación de la "Caja de Herramientas de Búsqueda de Algoritmos QSAR/QSPR" ("QSAR/QSPR Search Algorithms Toolbox") en la página de la comunidad de usuarios de MATLAB (MATLAB Central http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=19578&objectType=FILE). Este paquete consta de varios algoritmos junto con un instructivo y

un ejemplo que incluye una matriz para poder seguir el instructivo paso a paso. Se recomienda usar este mismo paquete antes del uso del resto de los algoritmos.

8 Apéndice

"La recompensa de una buena acción es haberla hecho" (Lucio A. Séneca)

8.1 Ejemplo de la diferencia entre RM y MRM

Para poder visualizar la diferencia entre MRM y RM los aplicaremos a la base de datos que hemos denominado FLUOR, que consiste en 116 compuestos orgánicos caracterizados por 1268 descriptores teóricos. Obtendremos el modelo oprimo de d=7 descriptores de el conjunto total. (D=1268)

Arbitrariamente se tomo un conjunto inicial **d**=$\{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$ cuya desviación estándar es $S_{(0)} = 0.771$ y seguiremos el camino 1 que lleva a los resultados que se pueden ver en la Figura 4.3.1.

La Tabla 8.1.1 muestra un resumen del procedimiento donde se puede observar el error relativo del coeficiente de la regresión para los descriptores y la constante de regresión C, asimismo se puede ver como S disminuye y R aumenta en cada paso del algoritmo.

En el camino 1 primero se cambia X_1; cada cambio se indicará con la notación (X_{viejo}, X_{nuevo})

De todas las 1261 $(D-d)$ variables, la sustitución que minimiza S es (X_1, X_{1068}) dando un $S(1)$= 0.689.

Ahora se prosigue reemplazando la variable con mayor error relativo X_6 con el resto de los 1261descriptores (X_{1068} ahora está fuera del conjunto total de descriptores y X_1 está dentro del mismo) y encontramos que la sustitución (X_6, X_{40}) produce la menor desviación estándar $S(2)$ = 0.634.

Ahora la variable con el mayor error relativo pasó a ser X_3. Luego de su reemplazo con todos los 1261 descriptores, se llega a que la sustitución (X_3, X_{411}) da el valor mínimo $S(3)$ = 0.602.

En el próximo paso la variable con mayor error relativo es X_7 y luego de su reemplazo por los 1261 descriptores, tenemos (X_7, X_{697}) y $S(4)$ = 0.574.

De todas las variables todavía no reemplazadas, X_2 es la que posee un mayor error relativo. El reemplazo con los 1261 descriptores no lleva a un modelo con menor S.

Hasta este punto MRM y RM tienen exactamente el mismo comportamiento. A partir de aquí la diferencia se hará visible.

Primero se mostrará como hubiese continuado RM. Como el reemplazo de X_2 no lleva a un modelo con menor S, X_2 permanece en su posición y no es reemplazado. Exactamente lo mismo ocurre con los próximos descriptores X_5 y X_4. Empezar el proceso nuevamente no lleva a un modelo con un S menor, por lo tanto el mejor modelo encontrado en este caso tiene una desviación estándar $S(4) = 0.574$.

Ahora continuaremos con MRM. Aún cuando el reemplazo del descriptor X_2 no lleva a un modelo con menor S, el descriptor es reemplazado de todas formas con el descriptor que da el menor S de los 1261 descriptores restantes; esto nos lleva a una sustitución (X_2, X_{1110}) con $S(5) = 0.580$. En este paso S ha aumentado levemente. Como se vera en los próximos pasos esto está lejos de ser un problema ya que luego un S aún más bajo será encontrado, mostrando que el incremento en S era necesario para salir de un mínimo.

En el próximo paso encontramos nuevamente que el reemplazo del descriptor que aún no fue reemplazado y que tiene el mayor error en el coeficiente (X_5) lleva a la sustitución (X_5, X_{394}) que produce un modelo con desviación estándar aún mayor $S(6) = 0.593$.

Sin embargo en el próximo paso el reemplazo de X_4 (el descriptor con mayor error en el coeficiente que permanece sin modificarse) por todos los 1261 descriptores conlleva a la sustitución (X_4, X_{1050}) que da un $S(7) = 0.545$ que es aún menor que el encontrado en el cuarto paso: $S(4) = 0.574$.

Al continuar el procedimiento, S continúa su tendencia a decrecer, como puede observarse en la Figura 4.3.1, en este caso llevando al menor valor luego de 222 pasos. El mejor modelo encontrado da un $S(222) = 0.4572$ y un R=0.9835, teniendo la forma:

$$\ln P = 0.065(\pm 0.3) - 3.9029(\pm 0.6)X_{425} - 0.0544(\pm 0.005)X_{240} - 0.063(\pm 0.004)X_{40} - 0.3749(\pm 0.01)X_{200} + 1.7051(\pm 0.2)X_{480} - 23.6913(\pm 2.7)X_{1095} + 0.008(\pm 0.001)X_{256}$$

Los descriptores moleculares que aparecen en la ecuación combinando aspectos bi y tridimensionales de la estructura molecular, y pueden clasificarse como Auto correlaciones 2D, cuatro descriptores Topológicos, un Índice de Aromaticidad y un

descriptor GETAWAY[119]. Los nombres de estos descriptores y su significado pude encontrarse en la Tabla 8.1.2.

Tabla 8.1.1 Evolución de MRM. Número de descriptores en el modelo con el correspondiente error relativo en los coeficientes de la regresión, *S* y *R* para cada paso del algoritmo. C significa constante de la regresión.

Paso Nº	Número de descriptor/ Error Relativo del coeficiente de Regresión								*S*	*R*
0	C	**1**	2	3	4	5	6	7	0.771	0.952
	28.29	90.12	38.95	59.59	20.36	194.94	84.91	50.21		
1	C	**1068**	2	3	4	5	6	7	0.689	0.962
	21.34	18.58	41.89	67.67	15.74	66.44	**796.66**	35.89		
2	C	1068	2	3	4	5	**40**	7	0.634	0.968
	15.62	16.69	31.43	**43.14**	10.58	35.56	22.60	27.24		
3	C	1068	2	**411**	4	5	40	7	0.602	0.971
	16.16	15.48	19.93	23.67	6.45	9.06	20.34	**82.96**		
4	C	1068	2	411	4	5	40	**697**	0.574	0.974
	8.76	17.92	**12.77**	18.99	5.29	7.07	18.49	28.74		
5	C	1068	**1110**	411	4	5	40	697	0.580	0.973
	9.24	23.11	13.12	18.97	6.49	**6.91**	21.53	23.96		
6	C	1068	1110	411	4	**394**	40	697	0.593	0.972
	6.50	26.58	10.69	16.79	**7.41**	7.14	23.55	15.56		
7	C	1068	204	411	**1050**	394	40	697	0.545	0.974
	45.13	**27.90**	25.83	21.09	4.84	5.85	13.76	12.49		
⋮										
222	C	425	240	40	200	480	1095	256	0.457	0.984
	414.91	16.10	8.35	6.05	3.37	10.05	11.34	12.30		

Tabla 8.1.2 Información de los descriptores del mejor modelo encontrado en el ejemplo

Descriptor			
Número	**Nombre**	**Tipo**	**Significado**
X_{425}	MATS1p	2D Autocorrelations	Moran autocorrelation - lag 1 / weighted by atomic polarizabilities
X_{240}	piPC03	Topological	Molecular multiple path count of order 03
X_{40}	IAC	Topological	Total information index of atomic composition
X_{200}	SEigv	Topological	Eigenvalue sum from van der Waals weighted distance matrix
X_{480}	AROM	Aromaticity indices	Aromaticity (trial)
X_{1095}	R3u+	GETAWAY	R maximal autocorrelation of lag 3 / unweighted
X_{256}	D/Dr10	Topological	Distance/detour ring index of order 10

8.2 Solución exacta en una base de datos modificada

A partir de una sugerencia de un colega se llevó a cabo la siguiente experiencia, la cual pude resultar útil en futuros trabajos donde se pretenda comparar algoritmos con soluciones exactas usando bases de datos grandes.

La prueba constó en forzar una solución exacta conocida para ver si los algoritmos eran capases de encontrarla. Para ello se seleccionaron siete descriptores al azar de una de las bases de datos (MES) y luego se calculó la propiedad correspondiente (ED50) usando estos descriptores. Se reemplazó la propiedad experimental por la calculada para estos descriptores, haciendo que de esta forma la desviación estándar *S* para este conjunto de descriptores fuera igual a cero; debido a que los valores predichos por estos siete descriptores pasaron a ser exactamente iguales a los supuestos valores experimentales. De esta forma se conoció de antemano el conjunto de descriptores del modelo que tenía el mínimo global en *S*.

Al hacer la búsqueda en esa base de datos modificada, tanto RM como ERM encontraron este conjunto de siete descriptores con S=0. Esto por un lado mostró la excelente calidad de ambos algoritmos, por el otro lado mostró que este nuevo problema con la base de datos alterada no permitía comparar entre ambos algoritmos, siendo esto último posible usando la base de datos original.

Por esta razon un segunda alternativa fue probada. En este caso lo que se hizo fue agregar un descriptor unico que combinado con otros seis elegidos al azar diera una propiedad calculada idéntica a la experimental; haciendo nuevamente S=0. Este descriptor fue anadido a la base de datos. Nuevamente tanto ERM como RM pudieron encontrar este conjunto de descriptores obteniendose exactamente los mismos resultados que con la primer alternativa.

Se supuso que la causa de la imposibilidad de comparar los algoritmos era debido a que el nuevo problema forzado estaba demasiado simplificado y se atribuyó esta simplificacion a que el nuevo minimo global era mucho menor que el verdadero minimo global desconocido. Por esta razón se llevaron a cabo distintas pruebas usando las dos alternativas antes mencionadas agregandoles un ruido que hiciera que *S* sea igual a 0.0001, 0.005, 0.01, 0.1 y 0.2. Los resultados mostraron que para el caso de S=0.0001 nuevamente tant o ERM como RM encontraron el conjunto de descriptores de menor *S*.

la solución. Para el resto de los casos la solucion encontrada fue aún menor a la seteada por lo que no habia certeza de que fuese el mínimo global.

Por lo tanto luego de estas pruebas se pudo llegar a la conclusión de que tanto ERM como RM tienen un excelente desempeño encontrando minimos globales y que no es posible alterar el problema en forma realística ya que al hacerlo se simplifica demasiado haciendo que ambos algoritmos enuentren el minimo global impidiendose de esta forma su comparación.

8.3 RM y AG combinados: pruebas no exitosas

8.3.1 Mutación orientada por *der*

El primer intento de usar un algoritmo híbrido entre AG y RM consistió en cambiar el operador de mutación aleatorio, es decir que cambia cualquiera de los *d* descriptores por otro descriptor elegido de manera aleatoria entre los restantes **D**, por uno que cambie el descriptor de mayor *der* (desviación estándar relativa).

Se probaron dos opciones, la primera fue elegir el nuevo descriptor probando uno a uno el resto de los descriptores en **D** y eligiendo aquel que diera menor *S*, este procedimiento es muy similar al usado en RM, y la opción se llamó *mutrm*. En la segunda opción que se denominó *mutder* más cercana a la filosofía de AG el nuevo descriptor fue elegido aleatoriamente.

Se hicieron pruebas con los parámetros estándar y con 100 individuos. En todos los casos los resultados mostraron ser peores que los del AG sin modificar; cabe además resaltar que en la primera opción el número de regresiones necesarias por cada generación aumenta considerablemente haciéndose un proceso muy lento.

En los siguientes gráficos se puede observar el comportamiento de los algoritmos usando las variaciones antes descriptas, en el caso del *mutrm* se observa que el algoritmo converge rápidamente cumpliendo la condición de que un individuo ocupe más del 90% de la población. Esto muestra que el comportamiento se asemeja el de RM; es decir de un algoritmo que tiene una convergencia muy veloz aunque con resultados más pobres que AG sin modificar. Para el caso de *mutder* el comportamiento es muy similar a GA, con la diferencia que la eficacia para encontrar mínimos en *S* es menor.

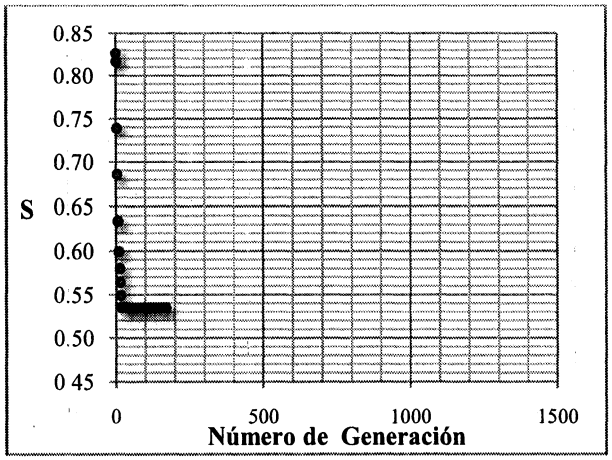

Figura 8.3.1 Desempeño de AG con ***mutrm*** para IND=20, GGAP=0.9, CrossP=0.6, MutP=0.7/*d*

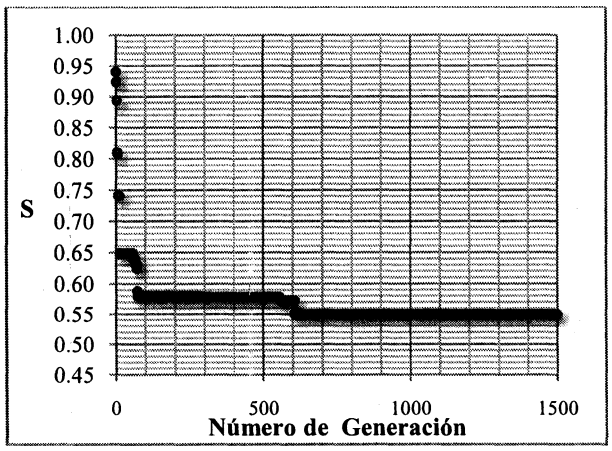

Figura 8.3.2 Desempeño de AG con ***mutder*** para IND=20, GGAP=0.9, CrossP=0.6, MutP=0.7/*d*

8.3.2 Entrecruzamiento orientado por *der*

El segundo intento fue cambiar el operador de entrecruzamiento aleatorio, el que toma dos individuos y entrecruza aleatoriamente alguno de sus cromosomas (descriptores), por uno que el entrecruzamiento lo haga de forma tal que uno de los nuevos individuos reciba el descriptor que tengan menor *der*. Al nuevo operador se lo llamó *crossder*.

Nuevamente se hicieron pruebas con los parámetros estándar y con 100 individuos y los resultados mostraron ser peores que los del AG original. En el gráfico se pude ver que el comportamiento no aparenta ser diferente al del AG original.

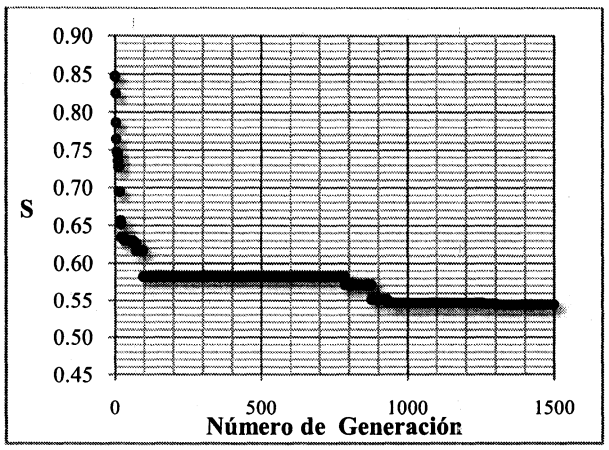

Figura 8.3.3 Desempeño de AG con ***crossder*** para IND=20, GGAP=0.9, CrossP=0.6, MutP=0.7/*d*

8.3.3 Mutación y Entrecruzamiento orientados por *der*

Por ultimo se combinaron los algoritmos antes presentados para verificar si la combinación de ambos podría llegar a ser mejor que los dos por separado.

Una vez más se hicieron pruebas con los parámetros estándares y con 100 individuos y los resultados mostraron ser peores que los del AG original. En el gráfico se observa que el algoritmo converge cumpliendo la condición de que un individuo ocupe más del 90% de la población, esto muestra que el comportamiento se aproxima al de RM, a pesar de que se usó el algoritmo de mutación mas parecido a AG (*mutder*).

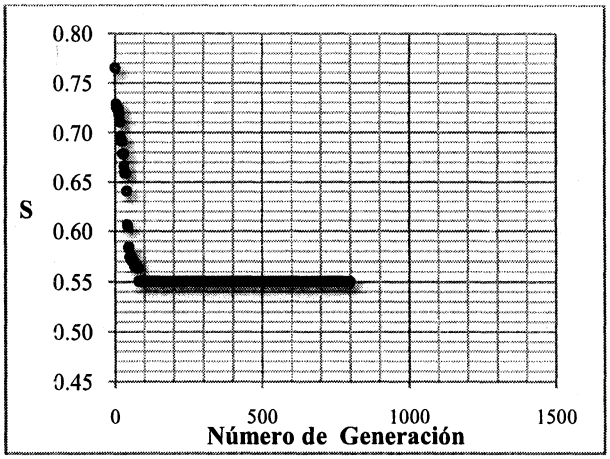

Figura 8.3.4 Desempeño de AG con ***crossder*** y ***mutder*** para IND=20, GGAP=0.9, CrossP=0.6, MutP=0.7/*d*

8.4 ERM con conjunto inicial de máximo S

Uno podría pensar que iniciar un algoritmo ERM con un punto de partida que tenga un valor muy bajo de *S* podría ser beneficioso para continuear disminuyendo la misma. Sin embargo esto pude favorecer a que el algoritmo quede trabado en este mínimo impidiendo que continúe la optimización.

Por esta razón se probaron soluciones de partida con un muy alto valor de *S*, para empezar el algoritmo lo más lejos posible de algún mínimo local de S y de esta forma tener más posibilidades de llegar al mínimo global.

La maximización de *S* se llevó a cabo usando dos opciones:

- Usando RM pero en sentido inverso, es decir en lugar de minimizando maximizando *S*, este algoritmo lo podríamos denominar RM_{inv}. Este algoritmo tiene la misma forma que RM con la salvedad de que en cada paso selecciona el conjunto con mayor S y se elije el descriptor con menor desviación estándar relativa (*der*) para seguir adelante.
- Un algoritmo de maximización de S basado en FSR[74] al cual podríamos denominar FSR_{inv}, el cual es simplemente una maximización de *S* usando una regresión "paso a paso".

Para comparar los resultados se usó adicionalmente un algoritmo que constaba de un ERM modificado de la siguiente forma: se tomaron conjuntos de descriptores con correlaciones máximas entre los mismos de 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 y 0.9

como puntos de partida de RM. Luego al mejor modelo encontrado se lo refinó con una secuencia MRM-RM adicional completando lo que sería equivalente a un el ciclo ERM. La elección de soluciones de partida fue realizada de este modo debido a que en el uso practico de RM se habían observado muy buenos resultados usando las mismas. Al intentar usar directamente las soluciones de partida con distintas correlaciones máximas en ERM no se vieron mejoras apreciables y el incremento computacional aumento en un factor de 10 por lo que esta opción se descartó.

Cabe mencionar que adicionalmente se estudió la relación entre soluciones de partida con distinto grado de correlación y los resultados de los modelos obtenidos (*S*) no encontrándose relación alguna entre las mismas.

En la Tabla 8.4.1 se volcaron los resultados incluyendo los mejores de la Tabla 4.3.2 de forma de además poder comparar los mismos con los obtenidos al usar puntos de partida aleatorios.

Al observar la tabla se pude ver que los mejores resultados corresponden a los puntos de partida aleatorios. Como en ese caso se usaron tres soluciones de partida distintas, esto indicaría que el uso de varias soluciones distintas da mejores resultados que al intentar alternativas más sofisticadas que inicialmente maximizan *S*. Esto estaría en línea con los resultados obtenidos para RM en la sección 4.5.2.

Al comparar los resultados obtenidos al usar RM_{inv} y FSR_{inv} se puede ver que el primero da mejores resultados lo que es probablemente por el hecho que al ser más eficiente RM encuentra una solución inicial de mayor *S*, que sería más lejana a cualquier mínimo local.

De los resultados se desprende la posibilidad de usar una nueva opción que sería una solución inicial con alto S usando RM_{inv} y otras aleatorias; esto junto con la comparación respecto a la misma cantidad de soluciones de partida aleatorias se verá en la próxima sección.

Tabla 8.4.1 Desviación estándar (S) encontrada usando distintos puntos de partida.

Base de Datos	Algoritmo			
	ERM	MRM-RM	ERM	ERM
	Punto de partida			
	Aleatorio	**RM (dif. sol.)**	**RM_{inv}**	**FSR_{inv}**
MES	**0.2896**	0.2950	0.2933	0.2950
GI	**0.4367**	0.4391	0.4421	0.4421
GABA	0.3961	0.4160	**0.3929**	**0.3929**
FLUOR	**0.4328**	0.4408	**0.4328**	0.4831

8.5 Algoritmos ERM y RM

A continuación se encuentra el código de los algoritmos más importantes que se usaron y/o desarrollaron durante el trabajo de Tesis. Para usar los mismos se deberán seguir los siguientes pasos:

- Tener instalado el MATLAB
- Copiar el código en archivos de texto
- Nombrar los archivos según indica los títulos de la sección (por ejemplo la el código de la sección **8.2.1 erm.m** deberá estar en un archivo denominado "erm.m"
- Asegurarse que la extensión de los archivos sean ".m" y no ".txt"
- Poner las funciones y sub-funciones en la misma carpeta o en carpetas reconocidas por MATLAB
- Usar las funciones como se hace normalmente en MATLAB
- Los comentarios aparecen antecedidos por un signo porcentaje (%) y figuran en color verde (como se mencionara se encuentran en ingles ya que el mismo fue necesario para su publicación online)

8.5.1 erm.m

```
function [VecTot, TOT, time] = erm(P, Vec, Mat)
```

```
%erm returns a matrix containing the best models for all the paths of
the Enhanced Replacement Method.
%TOT contains all the relative results of each step showing the
evolution of the method.
%
%       Input:
%              P              Property vector
%              Vec            Initial descriptors vector
%              Mat            Descriptors matrix with descriptors pool
%
%      Returns:
%
%             VecTot            vector containing the best model for all
the
%                                 paths of the Replacement Method
%             TOT               contains all the relative results
%                                 showing the evolution of the method.
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2007

TOT=[];
VecTot=[];
time=cputime;
warning off
[k_v,n_v]=size(Vec);

for k=1:n_v;

Sr=rms(P,Vec,Mat);
TOT(k).A(1,:)=[Sr,Vec];

VecA=rmsr(P, Vec, Mat, k);
Po(1)=k;
VecI=VecA;
if n_v==1
    VecTot=[1,VecI];
    TOT=VecI;
    time=cputime-time
    return
end
VecI(1)=[];
COEF=rmder(P,VecI,Mat);
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(2)=pos;
TOT(k).A(2,:)=VecA;
for i=2:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+1,:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
```

```
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end

for j=1:3;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end
end

for j=4:100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(2*n_v),:)
     break
   end

end

jj=j;
for j=jj:jj+100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rma2(P, VecI, Mat, pos);
    VecI=VecA;
```

```
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(4*n_v),:)
        Po=[];
        break
   end
end
VecQ=find(TOT(k).A==min(TOT(k).A(:,1)));
VecI=[TOT(k).A(VecQ(1),:)];
VecI(1)=[];
jjj=j;
for j=jjj:jjj+100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(2*n_v),:)
        Po=[];
        break
   end
end


VecP=find(TOT(k).A==min(TOT(k).A(:,1)));
VecTot(k,:)=[k,TOT(k).A(VecP(1),:)];
VecTot=sortrows(VecTot,2);
end
warning on
time=cputime-time
% end
```

8.5.2 rmt.m

```
function [VecTot, TOT, time] = erm(P, Vec, Mat)
```

```
%erm returns a matrix containing the best models for all the paths of
the Enhanced Replacement Method.
%TOT contains all the relative results of each step showing the
evolution of the method.
%
%       Input:
%               P               Property vector
%               Vec             Initial descriptors vector
%               Mat             Descriptors matrix with descriptors pool
%
%
%      Returns:
%
%             VecTot             vector containing the best model for all
the
%                                 paths of the Replacement Method
%             TOT                contains all the relative results
%                                 showing the evolution of the method.
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2007


TOT=[];
VecTot=[];
time=cputime;
warning off
[k_v,n_v]=size(Vec);

for k=1:n_v;


Sr=rms(P,Vec,Mat);
TOT(k).A(1,:)=[Sr,Vec];


VecA=rmsr(P, Vec, Mat, k);
Po(1)=k;
VecI=VecA;
if n_v==1
    VecTot=[1,VecI];
    TOT=VecI;
    time=cputime-time
    return
end
VecI(1)=[];
COEF=rmder(P,VecI,Mat);
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(2)=pos;
TOT(k).A(2,:)=VecA;
for i=2:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+1,:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
```

```
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end

for j=1:3;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end
end

for j=4:100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(2*n_v),:)
      break
   end

end

jj=j;
for j=jj:jj+100;
COER=COEF;
COER(Po)=[];
```

```
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rma2(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(4*n_v),:)
        Po=[];
        break
   end
end
VecQ=find(TOT(k).A==min(TOT(k).A(:,1)));
VecI=[TOT(k).A(VecQ(1),:)];
VecI(1)=[];
jjj=j;
for j=jjj:jjj+100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(2*n_v),:)
        Po=[];
        break
   end
end


VecP=find(TOT(k).A==min(TOT(k).A(:,1)));
VecTot(k,:)=[k,TOT(k).A(VecP(1),:)];
VecTot=sortrows(VecTot,2);
end
warning on
time=cputime-time
% end
```

8.5.3 stepwise.m

```
function [ResSW] = stepwise(P, Mat, MaxDesc)

% stepwise finds the solution with lower S_err for one descriptor,
then for
% two keeping the first and so on.
%      Input:
%             P             Property vector
%             Mat           Matrix with descriptors pool
%             MaxDesc       Maximum number of descriptors
%
%     Returns:
%
%             ResSW        Matrix with S_err and the number of the
%                           descriptors in the models found by the
stepwise method
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 7 March 2007

time=cputime;
warning off
if (nargin < 3)
   error('stepwise requires at least 3 input variables. Type ''help
stepwise''.');
end

[k, n_m] = size(Mat);

VecJ=[];
ResSW=[];

for i=1:MaxDesc
    Smin=1000000;   % A very big number compared to a normal S
necessary just to start the program
    for j=1:n_m;
    VecI=[VecJ,j];
    Ser=rms(P,VecI,Mat);
        if (Ser<Smin)
            Smin=Ser;
            VecK=VecI;
        end
    end
VecJ=VecK;
ResSW(i,1)=Smin;
    for k=2:MaxDesc+1
      [n_v m_v]=size(VecK);
        if k-1>m_v
        ResSW(i,k)=NaN;
        else
        ResSW(i,k)=VecK(k-1);
        end
    end
end
time=cputime-time
warning on

%End
```

8.5.4 lss.m

```
function Reslss = lss(P, Vec, Matr,TrainS)

% lss returns the same results as ls adding the standardized
coefficients in the last row of the matrix. Number of the descriptors
are in the
% first row (NaN represents the regression constant in the first
place, and Not
% Available in any other place.)
% In the second row, R, S, F, the average of the squared residuals,
AIC and
% FIT.
% In the third row the regression coefficients. In the fourth row the
errors in the coefficients.
% In the fifth row the relative errors in the coefficients deviation
of the regression coefficients.
% The sixth row contains the P-values.
% Final row contains the standardized coefficients.

%       Input:
%              P              Property vector
%              Vec            Descriptor number vector
%              Matr           Descriptors matrix with descriptors pool
%              TrainS         Training set
%
%
%      Returns:
%
%             Res            Multi-linear standarized regression result
matrix
%
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007

Y=P(TrainS);
Mat=Matr(TrainS,:);

if (nargin < 3)
   error('rmder requires at least 3 input variables. Type ''help
rmder''.');
end

% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
% Check that independent (X) and dependent (Y) data have compatible
dimensions
% ------------------------------------------------------------------
--------
```

```
     [n_x, k] = size(X);
[n_y,columns] = size(Y);

if n_x ~= n_y,
    error('The number of rows in Y must equal the number of rows in
Mat.');
end

if columns ~= 1,
    error('Y must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
%  ------------------------------------------------------------------
      X = [ ones(n,1) X ]   ;


if rcond(inv(X' * X)) < 1e-100
    S_err = 100;
    return;
end
       XTXI = inv(X' * X);
      Coefficients = XTXI * X' * Y ;


%  Calculate the fitted regression values
%  --------------------------------------

    Y_hat = X * Coefficients;

%  Calculate R-squared
%  -------------------
% The calculation used for R-squared and the F-statistic here are
based
% on the total, un-corrected sum of the squares as describe by Neter
and
% Myers. Note that the meaning of R-squared changes for the case of
% regression without a y-intercept. This approach will yield the same
% results as SysStat, SAS, SPSS and BMDP but will differ from that of
% Excel, Quattro Pro, and the MATLAB regress.m function (for the case
of
% no y-intercept in the model -- all packages including this one will
% agree for the case of linear regression with a
% y-intercept). Essentially, it is wise to find a way to
% keep the y-intercept (even if it is near zero) in the model to
analyze
% it in a meaningful way that everyone can understand.


   RSS = norm(Y_hat - mean(Y))^2;    % Regression sum of squares.
   TSS = norm(Y - mean(Y))^2;        % Total sum of squares (regression
plus residual).
   R_sq = RSS / TSS;                 % R-squared statistic.
   R_ = (R_sq)^0.5;                  % R.


% $$$ % Alternative calculation of R-squared
```

```
% $$$ % ========================================
% $$$ % The follwing equation is from Judge G, et al. "An Introduction
to the theory
% $$$ % and practice of econometrics", New York : Wiley, 1982. It is
the
% $$$ % squared (Pearson) correlation coefficient between the
predicted and
% $$$ % dependent variables. It is the same equation regardless of
whether an
% $$$ % intercept is included in the model; however, it may yield a
negative
% $$$ % R-squared for a particularily bad fit.
% $$$ covariance_Y_hat_and_Y = (Y_hat - mean(Y_hat))' * (Y - mean(Y));
% $$$ covariance_Y_hat_and_Y_hat = (Y_hat - mean(Y_hat))' * (Y_hat -
mean(Y_hat));
% $$$ covariance_Y_and_Y = (Y - mean(Y))' * (Y - mean(Y));
% $$$ R_sq = (covariance_Y_hat_and_Y / covariance_Y_hat_and_Y_hat) *
...
% $$$         (covariance_Y_hat_and_Y / covariance_Y_and_Y);

%  Calculate residuals and standard error
%  ---------------------------------------

    residuals = Y - Y_hat;

          S_err = sqrt(residuals' * residuals / (n - k - 1) );
          Ave_res= (residuals' * residuals)/n;

%  Calculate the standard deviation  for the regression coefficients
%  ------- ------------------------------------------------------------
----------
    covariance = XTXI .* S_err^2;

    C = sqrt(diag(covariance, 0));
    Er = abs(C./Coefficients)*100;

    % (n.b. Need to perform a 2-tailed t-test)
    % ***************************************

      p_value = 2 * (1 - tcdf(abs(Coefficients./C), (n - (k + 1))));


    Coef_stats = [ Coefficients, C, (Coefficients./C), p_value];

% Estimator of error variance.
% ----------------------------

     SSR_residuals = norm(Y - Y_hat)^2;
     TSS = norm(Y - mean(Y))^2;      % Total sum of squares (regression
plus residual).

     F_val = (TSS - SSR_residuals) / k / ( SSR_residuals / (n - (k +
1)));

     F_val = [F_val (1 - fcdf(F_val, k, (n - (k + 1)))) ];


  % AIC and FIT
```

```
  % ----------------------------

  AIC=sum(residuals.*residuals)*(n + k + 1)/(n-k-1)^2;

  FIT=R_^2*(n-k-1)/((n+k^2)*(1-R_^2));

% Standarization
% ----------------------------
Yst=Y-mean(Y);
S_y = sqrt(sum(Yst.*Yst)/(n - 1));
SRes_x=[];
for j =1:k+1
    Xst=X(:,j)-mean(X(:,j));
    S_x = sqrt(sum(Xst.*Xst)/(n - 1));
    SRes_x=[SRes_x;S_x];
end

CoefficientsStd=(Coefficients.*(SRes_x/S_y));

% Results
% ----------------------------

for i=1:k+1
    Res1(i)=NaN;
end

Res1(1)=R_;
Res1(2)=S_err;
Res1(3)=F_val(1);
Res1(4)=Ave_res;
Res1(5)=AIC;
Res1(6)=FIT;

[n_r, n_c] = size(Res1);

for i=1:n_c
    Vec1(i)=NaN;
   Coefficients1(i)=NaN;
    C1(i)=NaN;
    Er1(i)=NaN;
    p_value1(i)=NaN;
    CoefficientsStd1(i)=NaN;
end

for i=1:k+1
    Coefficients1(i)=Coefficients(i);
    C1(i)=C(i);
    Er1(i)=Er(i);
    p_value1(i)=p_value(i);
    CoefficientsStd1(i)=CoefficientsStd(i);
 end

 for i=1:k
    Vec1(i+1)=Vec(i);
 end

Reslss =[Vec1; Res1; Coefficients1;C1;Er1;p_value1;CoefficientsStd1];
%End
```

8.5.5 cx.m

```
function  [CorrCell] = cx(Vec, Matr, Names, TrainS)
%
%       or [CorrMat] = cx(ModMat)
%
%       Input:
%
%         Vec             Numbers of the descriptors in the model
%         Matr             Name of the matrix with the descriptors of the
pool
%         Names           Names of the descriptors in the pool
%
%    or   ModMat          Name of the matrix with the descriptors of the
model
%
%      Returns:
%
%            CorrMat                  Correletion Matrix
%
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 7 March 2007


[r_v c_v]=size(Vec);

if r_v==1
Mat=Matr(TrainS,:);
    ModMat = Mat(:,Vec);
    [m d]=size(ModMat);
    for i=1:d;
        DA=ModMat(:,i);
        for j=1:d;
           DB=ModMat(:,j);
           R = lrcx(DA, DB);
           CorrMat(i,j)=R;
        end
    end
    Name2 = Names(:,Vec);
    CorrCell = [Name2;num2cell(CorrMat)'];
    CorrCell=[[{''}, Name2];CorrCell];
    xlswrite('C:\Corr.xls',[CorrCell]);
end

if r_v>1
    ModMat = Vec;
    [m d]=size(ModMat);
    for i=1:d;
        DA=ModMat(:,i);
        for j=1:d;
           DB=ModMat(:,j);
           R = lrcx(DA, DB);
           CorrMat(i,j)=R;
        end
    end
    CorrCell=CorrMat;
    xlswrite('C:\Corr.xls',[CorrCell]);
end
```

8.5.6 Subfunctions

8.5.6.1 rma2.m

```
function [Res] = rma2(P, Vec, Mat, Pth)

% rma2 replaces all descriptors from Mat except the ones in the given
descriptor vector in the given position Pth and returns
%      and returns the vector with lower S_err, using multiple linear
regression analysis.
%      (The initial descriptor is changed even if it has the lower
S_err).
%
%      Input:
%             P             Property vector
%             Vec          Descriptor vector
%             Mat           Descriptor matrix
%             Pth           Path to follow
%
%     Returns:
%
%             Res           Vector containing in the first place the
%                           S_err, and afterwards the corresponding
descriptor vector
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007


if (nargin < 4)
   error('the function requires at least 4 input variables. Type
''help rma2''.');
end

[k, n_m] = size(Mat);

Num=[1:n_m];
Num(Vec)=[];


[k,n_n]=size(Num);

Smin=10000;   % A very big number compared to a normal S necessary
just to start the program
for j=1:n_n;
Vec(Pth)=Num(j);
Ser=rms(P,Vec,Mat);
    if (Ser<Smin)
         Smin=Ser;
        desc=Num(j);
    end
end
Vec(Pth)=desc;
Res=[Smin, Vec];
%End of rma2
```

8.5.6.2 rmsr.m

```
function [Res] = rmsr(P, Vec, Mat, Pth)

% rmsr replaces all descriptors from Mat (except the ones in the given
descriptor vector) in the given position Pth and returns
%       the vector with lower S_err, using multiple linear regression
analysis.
%
%
%       Input:
%               P               Property vector
%               Vec            Descriptor vector
%               Mat             Descriptor matrix
%               Pth             Path to follow
%
%      Returns:
%
%               Res             Vector containing in the first place the
%                               S_err, and afterwards the corresponding
descriptor vector
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007


if (nargin < 4)
   error('rmsr requires at least 4 input variables. Type ''help
rmsr''.');
end

[k, n_m] = size(Mat);

Num=[1:n_m];
Vec2=Vec;
Vec2(Pth)=[];
Num(Vec2)=[];


[k,n_n]=size(Num);

Smin=1000000;    % A very big number compared to a normal S necessary
just to start the program
for j=1:n_n;
Vec(Pth)=Num(j);
Ser=rms(P,Vec,Mat);
    if (Ser<Smin)
         Smin=Ser;
        desc=Num(j);
    end
end
Vec(Pth)=desc;
Res=[Smin, Vec];
%End of rmsr
```

8.5.6.3 rms.m

```
function  [S_err] = rms(P, Vec, Mat)

% rms evaluates selection of descriptors using multiple linear
regression analysis.
%
%
%       Input:
%               P               Property vector
%               Vec             Descriptors vector
%               Mat             Descriptors matrix
%
%      Returns:
%
%               S_err             standard error
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007


if (nargin < 3)
   error('rms requires at least 3 input variables. Type ''help
rms''.');
end


% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
% Check that independent (X) and dependent (Y) data have compatible
dimensions
% ------------------------------------------------------------------
--------
     [n_x, k] = size(X);
[n_y,columns] = size(P);

if n_x ~= n_y,
    error('The number of rows in P must equal the number of rows in
Mat.');
end

if columns ~= 1,
    error('P must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
%  ----------------------------------------------------------------
```

```
      X = [ ones(n,1) X ]    ;

  XTXI = X' * X;
if rcond(XTXI) < 1e-25
    S_err = 10000000;
    return;
end

 XTXI = inv(XTXI);
       Coefficients = XTXI * X' * P ;


%  Calculate the fitted regression values
%  --------------------------------------

    P_hat = X * Coefficients;



%  Calculate residuals and standard error
%  --------------------------------------

    residuals = P - P_hat;

     S_err = sqrt(residuals' * residuals / (n - k - 1) );

%Edn
```

8.5.6.4 rmder.m

```
function [COEF] = rmder(P, Vec, Mat)

% rmder returns a vector containing the relative standard deviation of
the regression coefficients. The
%                              constant of the regression is not
included
%
%
%       Input:
%              P              Property vector
%              Vec            Descriptors vector
%              Mat            Descriptors matrix
%
%
%     Returns:
%
%             COEF            Vector containing the standard deviation
for
%                              the regression coefficients without the
constant
%
%
%
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007
```

```
if (nargin < 3)
   error('rmder requires at least 3 input variables. Type ''help
rmder''.');
end


% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
% Check that independent (X) and dependent (P) data have compatible
dimensions
% ------------------------------------------------------------------
--------
     [n_x, k] = size(X);
[n_y,columns] = size(P);

if n_x ~= n_y,
    error('The number of rows in P must equal the number of rows in
Mat.');
end

if columns ~= 1,
    error('P must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
%  ------------------------------------------------------------------
     X = [ ones(n,1) X ]   ;


XTXI = X' * X;
if rcond(XTXI) < 1e-25
    S_err = 10000000;
    return;
end

 XTXI = inv(XTXI);
 Coefficients = XTXI * X' * P ;


%  Calculate the fitted regression values
%  --------------------------------------

    P_hat = X * Coefficients;



%  Calculate residuals and standard error
%  --------------------------------------
```

```
    residuals = P - P_hat;

          S_err = sqrt(residuals' * residuals / (n - k - 1) );

%  Calculate the standard deviation  for the regression coefficients
%  ------------------------------------------------------------------
----------

    covariance = XTXI .* S_err^2;

    C = sqrt(diag(covariance, 0));
    Er = abs(C./Coefficients)*100;



COEF=[[Er]'];
COEF(1)=[];

% End
```

8.5.6.5 ld.m

```
function  [lindep] = ld(P, Vec, Mat)

% ld tests the lineal dependence of the descriptors in Vec including
the
% regression constant
%

%   Input:
%             P             Property vector
%             Vec           Initial descriptors vector
%             Mat           Descriptors matrix with descriptors pool
%
%
%     Returns:
%
%            lindep            if there is a linear dependence lindep
= 100
%                               otherwise lindep = 0

% Pablo R. Duchowicz; Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2007


if (nargin < 3)
   error('ld requires at least 3 input variables. Type ''help ld''.');
end

% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
```

```
% Check that independent (X) and dependent (P) data have compatible
dimensions
% ------------------------------------------------------------------------
--------
     [n_x, k] = size(X);
[n_y,columns] = size(P);

if n_x ~= n_y,
    error('The number of rows in P must equal the number of rows in
Mat.');
end

if columns ~= 1,
    error('P must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
%  ------------------------------------------------------------------------
      X = [ ones(n,1) X ]    ;



lindep=0;

nu=k+1;

if (rank(X)<nu)
    lindep = 100;
end;
%End
```

8.6 Algoritmos genéticos para QSAR/QSPR

8.6.1 sgaqsar.m

```
function [Best BChrom] = sgaqsar(NInd, NDesc, MAXGEN, GGAP,CrssP,
MutP, ExpDat, DescMat);

%
% This script implements the Simple Genetic Algorithm modified for
QSAR,
%  the unmodified algorithm is described
% in the examples section of the GA Toolbox manual.
%

% Author: Andrew G. Mercader, INIFTA, 2007
% Based on the initial file from: Andrew Chipperfield (23-Mar-94)

% Examples:
%
% NInd = 20;            % Number of individuals per subpopulations
% NDesc = 7;            % Number of descriptors in an individual
% MAXGEN = 1000;           % maximum Number of generations
% GGAP = 0.9;           % Generation gap, how many new individuals are
created
% ExpDat = P;           % Name of the vector with experimental data
% DescMat = Tsub;           % Name of the matrix with the descriptors
% CrssP = 0.6                %         Crossover Probability
% MutP = 0.7                        %Mutation Probability/Number of
Descriptors

warning off;
 [n_p,NTot]=size(DescMat);

% Build field descriptor
    FieldD = [ones(1,NDesc)*NTot];

% Initialise population
  Chrom = crtip(NInd, NDesc, DescMat);

% Reset counters
   Best = NaN*ones(MAXGEN,NDesc+1); % best in current population
   gen = 0;          % generational counter

% Evaluate initial population
   ObjVal = objfunq(ExpDat, Chrom, DescMat);
   BChrom=[ObjVal,Chrom];
   [C,I] = min(BChrom(:,1));

% Track best individual and display convergence
   Best(gen+1,:) = BChrom(I,:);
   plot((Best(:,1)),'ro');xlabel('generation'); ylabel('S');
   text(0.5,0.90,['Best = ',
num2str(Best(gen+1))],'Units','normalized');
   drawnow;


% Generational loop
   while gen < MAXGEN ,
```

```
    % Assign fitness-value to entire population
       FitnV = ranking(ObjVal);

    % Select individuals for breeding
       SelCh = select('sus', Chrom, FitnV, GGAP);

    % Recombine selected individuals (crossover)
       SelCh = recombin('xovsp',SelCh,CrssP);

        % Perform mutation on offspring
       SelCh = mutq(SelCh, MutP/NDesc, FieldD);

    % Evaluate offspring, call objective function
       ObjVSel = objfunq(ExpDat, SelCh, DescMat);

    % Reinsert offspring into current population
       [Chrom ObjVal]=reins(Chrom,SelCh,1,1,ObjVal,ObjVSel);
        BChrom=[ObjVal,Chrom];
        [C,I] = min(BChrom(:,1));
    % Increment generational counter
       gen = gen+1;

    % Update display and record current best individual
       Best(gen+1,:) = BChrom(I,:);
       plot((Best(:,1)),'ro'); xlabel('generation'); ylabel('S');
       text(0.5,0.90,['Best = ',
num2str(Best(gen+1))],'Units','normalized');
       drawnow;

       if max(percentq(Best(:,1))) > 90 && gen > 100,
           break
       end
  warning on;
   end
% End of GA
```

8.6.2 Sub-funciones

8.6.2.1 crtip.m

```
% CRTIP.M         (CReaTe an initial (Integer) Population)
%
% This function creates a population of given size of random Integer-
values.
% That are not linar dependant
%
% Syntax:       Chrom = crtip(Nind, NDesc, NTot);
%
% Input parameters:
%    Nind       - A scalar containing the number of individuals in the
new
%                 population.
%
%    NDesc    - A scalar containing the number of descriptors in an
%                individual.
```

```
%    Mat            Name of the matrix with the pool of descriptors
%
% Output parameter:
%    Chrom      - A matrix containing the random valued individuals of
the
%                 new population of size Nind by NDesc.

% Author:     Andrew G. Mercader based on a previous version by
Hartmut Pohlheim
% History:    23.11.93     file created (Hartmut Pohlheim)
%             25.02.94     clean up, check parameter consistency
(Hartmut
%             Pohlheim)
%             30.01.2007   QSAR Adaptation


function Chrom = crtip(Nind, NDesc, Mat);

% Check parameter consistency
   if nargin < 3, error('parameter missing'); end

   [mN, nN] = size(Nind);
   [mO, nP] = size(NDesc);
   [mQ, nR] = size(Mat);
NTot=nR;


   if (mN ~= 1 & nN ~= 1), error('Nind has to be a scalar'); end
   if (mO ~= 1 & nP ~= 1), error('NDesc has to be a scalar'); end


% Create initial population
% Each row contains one individual, the values of each variable
uniformly
% distributed between lower and upper bound (given by FieldDR)

Chrom=[];
for i=1:Nind
     for j=1:10000000
     Chrom1 = randint(1,NDesc,[1, NTot]);
     Corr=cxmax(Chrom1, Mat);
        if Corr < 0.99
            break
        end
     end
    Chrom=[Chrom;Chrom1];
   end

% End of function
```

8.6.2.2 objfunnq.m

```
% objfunq.m      (OBJective function for QSAR)
%
% This function implements a multiple linear regression for each
individual in the population returning S_err.
```

```
%
% Syntax:  ObjVal = objfunq(Y, Pob, Mat)
%
% Input parameters:
%
%              Y              Property vector
%              Pob            Population matrix (normaly named Chrom)
%              Mat            Descriptor matrix
%
% Output parameters:
%             ObjVal          Column vector containing the objective
values of the
%                             individuals in the current population.
%
%

% Author:      Andrew G. Mercader
%              INIFTA, La Plata, Argenitna
% History:     31.01.07     file created
%

function ObjVal = objfunq(Y, Pob, Mat);

if (nargin < 3)
   error('objfunq requires at least 3 input variables. Type ''help
objfunq''.');
end

% Compute population parameters
   [Nin,Nvar] = size(Pob);

for i=1:Nin;
    ObjVal(i,1)=funqsar(Y, i, Pob, Mat);
end
% End of function
```

8.6.2.3 ranking.m

```
% RANKING.M       (RANK-based fitness assignment)
%
% This function performs ranking of individuals.
%
% Syntax:  FitnV = ranking(ObjV, RFun, SUBPOP)
%
% This function ranks individuals represented by their associated
% cost, to be *minimized*, and returns a column vector FitnV
% containing the corresponding individual fitnesses. For multiple
% subpopulations the ranking is performed separately for each
% subpopulation.
%
% Input parameters:
%    ObjV       - Column vector containing the objective values of the
%                 individuals in the current population (cost values).
%    RFun       - (optional) If RFun is a scalar in [1, 2] linear
ranking is
%                 assumed and the scalar indicates the selective
pressure.
%                 If RFun is a 2 element vector:
%                 RFun(1): SP - scalar indicating the selective
pressure
```

```
%                RFun(2): RM - ranking method
%                         RM = 0: linear ranking
%                         RM = 1: non-linear ranking
%                If RFun is a vector with length(Rfun) > 2 it contains
%                the fitness to be assigned to each rank. It should
have
%                the same length as ObjV. Usually RFun is monotonously
%                increasing.
%                If RFun is omitted or NaN, linear ranking
%                and a selective pressure of 2 are assumed.
%    SUBPOP    - (optional) Number of subpopulations
%                if omitted or NaN, 1 subpopulation is assumed
%
% Output parameters:
%    FitnV     - Column vector containing the fitness values of the
%                individuals in the current population.
%

% Author:     Hartmut Pohlheim (Carlos Fonseca)
% History:    01.03.94     non-linear ranking
%             10.03.94     multiple populations

function FitnV = ranking(ObjV, RFun, SUBPOP);

% Identify the vector size (Nind)
   [Nind,ans] = size(ObjV);

   if nargin < 2, RFun = []; end
   if nargin > 1, if isnan(RFun), RFun = []; end, end
   if prod(size(RFun)) == 2,
      if RFun(2) == 1, NonLin = 1;
      elseif RFun(2) == 0, NonLin = 0;
      else error('Parameter for ranking method must be 0 or 1'); end
      RFun = RFun(1);
      if isnan(RFun), RFun = 2; end
   elseif prod(size(RFun)) > 2,
      if prod(size(RFun)) ~= Nind, error('ObjV and RFun disagree');
end
   end

   if nargin < 3, SUBPOP = 1; end
   if nargin > 2,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (Nind/SUBPOP) ~= fix(Nind/SUBPOP), error('ObjV and SUBPOP
disagree'); end
   Nind = Nind/SUBPOP;  % Compute number of individuals per
subpopulation

% Check ranking function and use default values if necessary
   if isempty(RFun),
      % linear ranking with selective pressure 2
         RFun = 2*[0:Nind-1]'/(Nind-1);
   elseif prod(size(RFun)) == 1
      if NonLin == 1,
         % non-linear ranking
```

```
        if RFun(1) < 1, error('Selective pressure must be greater
than 1');
        elseif RFun(1) > Nind-2, error('Selective pressure too big');
end
        Root1 = roots([RFun(1)-Nind [RFun(1)*ones(1,Nind-1)]]);
        RFun = (abs(Root1(1)) * ones(Nind,1)) .^ [(0:Nind-1)'];
        RFun = RFun / sum(RFun) * Nind;
     else
        % linear ranking with SP between 1 and 2
        if (RFun(1) < 1 | RFun(1) > 2),
           error('Selective pressure for linear ranking must be
between 1 and 2');
        end
        RFun = 2-RFun + 2*(RFun-1)*[0:Nind-1]'/(Nind-1);
     end
  end;

  FitnV = [];

% loop over all subpopulations
for irun = 1:SUBPOP,
  % Copy objective values of actual subpopulation
     ObjVSub = ObjV((irun-1)*Nind+1:irun*Nind);
  % Sort does not handle NaN values as required. So, find those...
     NaNix = isnan(ObjVSub);
     Validix = find(~NaNix);
  % ... and sort only numeric values (smaller is better).
     [ans,ix] = sort(-ObjVSub(Validix));

  % Now build indexing vector assuming NaN are worse than numbers,
  % (including Inf!)...
     ix = [find(NaNix) ; Validix(ix)];
  % ... and obtain a sorted version of ObjV
     Sorted = ObjVSub(ix);

  % Assign fitness according to RFun.
     i = 1;
     FitnVSub = zeros(Nind,1);
     for j = [find(Sorted(1:Nind-1) ~= Sorted(2:Nind)); Nind]',
        FitnVSub(i:j) = sum(RFun(i:j)) * ones(j-i+1,1) / (j-i+1);
        i =j+1;
     end

  % Finally, return unsorted vector.
     [ans,uix] = sort(ix);
     FitnVSub = FitnVSub(uix);

  % Add FitnVSub to FitnV
     FitnV = [FitnV; FitnVSub];
end


% End of function
```

8.6.2.4 select.m

```
% SELECT.M          (universal SELECTion)
```

```
%
% This function performs universal selection. The function handles
% multiple populations and calls the low level selection function
% for the actual selection process.

%
% Syntax:  SelCh = select(SEL_F, Chrom, FitnV, GGAP, SUBPOP)
%
% Input parameters:
%    SEL_F      - Name of the selection function
%    Chrom      - Matrix containing the individuals (parents) of the
current
%                 population. Each row corresponds to one individual.
%    FitnV      - Column vector containing the fitness values of the
%                 individuals in the population.
%    GGAP       - (optional) Rate of individuals to be selected
%                 if omitted 1.0 is assumed
%    SUBPOP     - (optional) Number of subpopulations
%                 if omitted 1 subpopulation is assumed
%
% Output parameters:
%    SelCh      - Matrix containing the selected individuals.

% Author:     Hartmut Pohlheim
% History:    10.03.94     file created

function SelCh = select(SEL_F, Chrom, FitnV, GGAP, SUBPOP);

% Check parameter consistency
   if nargin < 3, error('Not enough input parameter'); end

   % Identify the population size (Nind)
   [NindCh,Nvar] = size(Chrom);
   [NindF,VarF] = size(FitnV);
   if NindCh ~= NindF, error('Chrom and FitnV disagree'); end
   if VarF ~= 1, error('FitnV must be a column vector'); end

   if nargin < 5, SUBPOP = 1; end
   if nargin > 4,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (NindCh/SUBPOP) ~= fix(NindCh/SUBPOP), error('Chrom and SUBPOP
disagree'); end
   Nind = NindCh/SUBPOP;  % Compute number of individuals per
subpopulation

   if nargin < 4, GGAP = 1; end
   if nargin > 3,
      if isempty(GGAP), GGAP = 1;
      elseif isnan(GGAP), GGAP = 1;
      elseif length(GGAP) ~= 1, error('GGAP must be a scalar');
      elseif (GGAP < 0), error('GGAP must be a scalar bigger than 0');
end
   end

% Compute number of new individuals (to select)
```

```
   NSel=max(floor(Nind*GGAP+.5),2);

% Select individuals from population
   SelCh = [];
   for irun = 1:SUBPOP,
      FitnVSub = FitnV((irun-1)*Nind+1:irun*Nind); % Only works for
subpopulations
      ChrIx=feval(SEL_F, FitnVSub, NSel)+(irun-1)*Nind;
      SelCh=[SelCh; Chrom(ChrIx,:)];
   end


% End of function
```

8.6.2.5 recombin.m

```
% RECOMBIN.M       (RECOMBINation high-level function)
%
% This function performs recombination between pairs of individuals
% and returns the new individuals after mating. The function handles
% multiple populations and calls the low-level recombination function
% for the actual recombination process.
%
% Syntax:  NewChrom = recombin(REC_F, OldChrom, RecOpt, SUBPOP)
%
% Input parameters:
%    REC_F      - String containing the name of the recombination or
%                 crossover function
%    Chrom      - Matrix containing the chromosomes of the old
%                 population. Each line corresponds to one individual
%    RecOpt     - (optional) Scalar containing the probability of
%                 recombination/crossover occurring between pairs
%                 of individuals.
%                 if omitted or NaN, 1 is assumed
%    SUBPOP     - (optional) Number of subpopulations
%                 if omitted or NaN, 1 subpopulation is assumed
%
% Output parameter:
%    NewChrom   - Matrix containing the chromosomes of the population
%                 after recombination in the same format as OldChrom.

%  Author:     Hartmut Pohlheim
%  History:    18.03.94      file created


function NewChrom = recombin(REC_F, Chrom, RecOpt, SUBPOP);


% Check parameter consistency
   if nargin < 2, error('Not enough input parameter'); end

   % Identify the population size (Nind)
   [Nind,Nvar] = size(Chrom);

   if nargin < 4, SUBPOP = 1; end
   if nargin > 3,
```

```
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (Nind/SUBPOP) ~= fix(Nind/SUBPOP), error('Chrom and SUBPOP
disagree'); end
   Nind = Nind/SUBPOP;  % Compute number of individuals per
subpopulation

   if nargin < 3, RecOpt = 0.7; end
   if nargin > 2,
      if isempty(RecOpt), RecOpt = 0.7;
      elseif isnan(RecOpt), RecOpt = 0.7;
      elseif length(RecOpt) ~= 1, error('RecOpt must be a scalar');
      elseif (RecOpt < 0 | RecOpt > 1), error('RecOpt must be a scalar
in [0, 1]'); end
   end


% Select individuals of one subpopulation and call low level function
   NewChrom = [];
   for irun = 1:SUBPOP,
      ChromSub = Chrom((irun-1)*Nind+1:irun*Nind,:);
      NewChromSub = feval(REC_F, ChromSub, RecOpt);
      NewChrom=[NewChrom; NewChromSub];
   end


% End of function
```

8.6.2.6 mutq.m

```
% MUTQ.m (mutation for QSAR)
%
% This function takes the representation of the current population,
% mutates each element with given probability and returns the
resulting
% population.
%
% Syntax:   NewChrom = mut(OldChrom,Pm,BaseV)
%
% Input parameters:
%
%       OldChrom - A matrix containing the chromosomes of the
%              current population. Each row corresponds to
%              an individuals string representation.
%
%       Pm    - Mutation probability (scalar). Default value
%              of Pm = 0.7/Lind, where Lind is the chromosome
%              length is assumed if omitted.
%
%       BaseV     - Optional row vector of the same length as the
%              chromosome structure defining the base of the
%              individual elements of the chromosome. Binary
%              representation is assumed if omitted.
%
% Output parameter:
```

```
%
%       NewChrom - A Matrix containing a mutated version of
%              OldChrom.
%

% Author: Author:      Andrew G. Mercader based on a previous version
by Andrew Chipperfield
% Date: 31.01.07      file created

function NewChrom = mutq(OldChrom,Pm,BaseV)

% get population size (Nind) and chromosome length (Lind)
[Nind, Lind] = size(OldChrom) ;

% check input parameters
if nargin < 2, Pm = 0.7/Lind ; end
if isnan(Pm), Pm = 0.7/Lind; end

if (nargin < 3), BaseV = crtbase(Lind);  end
if (isnan(BaseV)), BaseV = crtbase(Lind);  end
if (isempty(BaseV)), BaseV = crtbase(Lind);  end

if (nargin == 3) & (Lind ~= length(BaseV))
   error('OldChrom and BaseV are incompatible'), end

% create mutation mask matrix
BaseM = BaseV(ones(Nind,1),:) ;

% perform mutation on chromosome structure
NewChrom =
rem(OldChrom+(rand(Nind,Lind)<Pm).*ceil(rand(Nind,Lind).*(BaseM-
1)),BaseM);
n=(Nind*Lind);
for i=1:n
    if NewChrom(i) == 0
       NewChrom(i) = 2;
    end
end
```

8.6.2.7 objfunq.m

```
% objfunq.m       (OBJective function for QSAR)
%
% This function implements a multiple linear regression for each
individual in the population returning S_err.
%
% Syntax:  ObjVal = objfunq(Y, Pob, Mat)
%
% Input parameters:
%
%              Y              Property vector
%              Pob            Population matrix (normaly named Chrom)
%              Mat            Descriptor matrix
%
% Output parameters:
%            ObjVal           Column vector containing the objective
values of the
%                             individuals in the current population.
```

```
%
%

% Author:       Andrew G. Mercader
%               INIFTA, La Plata, Argenitna
% History:      31.01.07      file created
%

function ObjVal = objfunq(Y, Pob, Mat);

if (nargin < 3)
   error('objfunq requires at least 3 input variables. Type ''help
objfunq''.');
end

% Compute population parameters
   [Nin,Nvar] = size(Pob);

for i=1:Nin;
    ObjVal(i,1)=funqsar(Y, i, Pob, Mat);
end
% End of function
```

8.6.2.8 reins.m

```
% REINS.M         (RE-INSertion of offspring in population replacing
parents)
%
% This function reinserts offspring in the population.
%
% Syntax: [Chrom, ObjVCh] = reins(Chrom, SelCh, SUBPOP, InsOpt,
ObjVCh, ObjVSel)
%
% Input parameters:
%    Chrom      - Matrix containing the individuals (parents) of the
current
%                 population. Each row corresponds to one individual.
%    SelCh      - Matrix containing the offspring of the current
%                 population. Each row corresponds to one individual.
%    SUBPOP     - (optional) Number of subpopulations
%                 if omitted or NaN, 1 subpopulation is assumed
%    InsOpt     - (optional) Vector containing the insertion method
parameters
%                 ExOpt(1): Select - number indicating kind of
insertion
%                           0 - uniform insertion
%                           1 - fitness-based insertion
%                           if omitted or NaN, 0 is assumed
%                 ExOpt(2): INSR - Rate of offspring to be inserted per
%                           subpopulation (% of subpopulation)
%                           if omitted or NaN, 1.0 (100%) is assumed
%    ObjVCh     - (optional) Column vector containing the objective
values
%                 of the individuals (parents - Chrom) in the current
%                 population, needed for fitness-based insertion
%                 saves recalculation of objective values for
population
```

```
%    ObjVSel    - (optional) Column vector containing the objective
values
%                 of the offspring (SelCh) in the current population,
needed for
%                 partial insertion of offspring,
%                 saves recalculation of objective values for
population
%
% Output parameters:
%    Chrom      - Matrix containing the individuals of the current
%                 population after reinsertion.
%    ObjVCh     - if ObjVCh and ObjVSel are input parameter, than
column
%                 vector containing the objective values of the
individuals
%                 of the current generation after reinsertion.

% Author:      Hartmut Pohlheim
% History:     10.03.94     file created
%              19.03.94     parameter checking improved

function [Chrom, ObjVCh] = reins(Chrom, SelCh, SUBPOP, InsOpt, ObjVCh,
ObjVSel);


% Check parameter consistency
   if nargin < 2, error('Not enough input parameter'); end
   if (nargout == 2 & nargin < 6), error('Input parameter missing:
ObjVCh and/or ObjVSel'); end

   [NindP, NvarP] = size(Chrom);
   [NindO, NvarO] = size(SelCh);

   if nargin == 2, SUBPOP = 1; end
   if nargin > 2,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (NindP/SUBPOP) ~= fix(NindP/SUBPOP), error('Chrom and SUBPOP
disagree'); end
   if (NindO/SUBPOP) ~= fix(NindO/SUBPOP), error('SelCh and SUBPOP
disagree'); end
   NIND = NindP/SUBPOP;  % Compute number of individuals per
subpopulation
   NSEL = NindO/SUBPOP;  % Compute number of offspring per
subpopulation

   IsObjVCh = 0; IsObjVSel = 0;
   if nargin > 4,
      [mO, nO] = size(ObjVCh);
      if nO ~= 1, error('ObjVCh must be a column vector'); end
      if NindP ~= mO, error('Chrom and ObjVCh disagree'); end
      IsObjVCh = 1;
   end
   if nargin > 5,
      [mO, nO] = size(ObjVSel);
      if nO ~= 1, error('ObjVSel must be a column vector'); end
```

```
      if NindO ~= mO, error('SelCh and ObjVSel disagree'); end
      IsObjVSel = 1;
   end

   if nargin < 4, INSR = 1.0; Select = 0; end
   if nargin >= 4,
      if isempty(InsOpt), INSR = 1.0; Select = 0;
      elseif isnan(InsOpt), INSR = 1.0; Select = 0;
      else
         INSR = NaN; Select = NaN;
         if (length(InsOpt) > 2), error('Parameter InsOpt too long');
end
         if (length(InsOpt) >= 1), Select = InsOpt(1); end
         if (length(InsOpt) >= 2), INSR = InsOpt(2); end
         if isnan(Select), Select = 0; end
         if isnan(INSR), INSR =1.0; end
      end
   end

   if (INSR < 0 | INSR > 1), error('Parameter for insertion rate must
be a scalar in [0, 1]'); end
   if (INSR < 1 & IsObjVSel ~= 1), error('For selection of offspring
ObjVSel is needed'); end
   if (Select ~= 0 & Select ~= 1), error('Parameter for selection
method must be 0 or 1'); end
   if (Select == 1 & IsObjVCh == 0), error('ObjVCh for fitness-based
exchange needed'); end

   if INSR == 0, return; end
   NIns = min(max(floor(INSR*NSEL+.5),1),NIND);    % Number of
offspring to insert

% perform insertion for each subpopulation
   for irun = 1:SUBPOP,
      % Calculate positions in old subpopulation, where offspring are
inserted
         if Select == 1,     % fitness-based reinsertion
            [Dummy, ChIx] = sort(-ObjVCh((irun-1)*NIND+1:irun*NIND));
         else                % uniform reinsertion
            [Dummy, ChIx] = sort(rand(NIND,1));
         end
         PopIx = ChIx((1:NIns)')+ (irun-1)*NIND;
      % Calculate position of Nins-% best offspring
         if (NIns < NSEL),  % select best offspring
            [Dummy,OffIx] = sort(ObjVSel((irun-1)*NSEL+1:irun*NSEL));
         else
            OffIx = (1:NIns)';
         end
         SelIx = OffIx((1:NIns)')+(irun-1)*NSEL;
      % Insert offspring in subpopulation -> new subpopulation
         Chrom(PopIx,:) = SelCh(SelIx,:);
         if (IsObjVCh == 1 & IsObjVSel == 1), ObjVCh(PopIx) =
ObjVSel(SelIx); end
   end


% End of function
```

8.6.2.9 sus.m

```
% SUS.M          (Stochastic Universal Sampling)
%
% This function performs selection with STOCHASTIC UNIVERSAL SAMPLING.
%
% Syntax:  NewChrIx = sus(FitnV, Nsel)
%
% Input parameters:
%    FitnV      - Column vector containing the fitness values of the
%                 individuals in the population.
%    Nsel       - number of individuals to be selected
%
% Output parameters:
%    NewChrIx   - column vector containing the indexes of the selected
%                 individuals relative to the original population,
shuffled.
%                 The new population, ready for mating, can be obtained
%                 by calculating OldChrom(NewChrIx,:).

% Author:      Hartmut Pohlheim (Carlos Fonseca)
% History:     12.12.93     file created
%              22.02.94     clean up, comments


function NewChrIx = sus(FitnV,Nsel);

% Identify the population size (Nind)
   [Nind,ans] = size(FitnV);

% Perform stochastic universal sampling
   cumfit = cumsum(FitnV);
   trials = cumfit(Nind) / Nsel * (rand + (0:Nsel-1)');
   Mf = cumfit(:, ones(1, Nsel));
   Mt = trials(:, ones(1, Nind))';
   [NewChrIx, ans] = find(Mt < Mf & [ zeros(1, Nsel); Mf(1:Nind-1, :)
] <= Mt);

% Shuffle new population
   [ans, shuf] = sort(rand(Nsel, 1));
   NewChrIx = NewChrIx(shuf);


% End of function
```

8.6.2.10 xovsp.m

```
% XOVSP.M         (CROSSOVer Single-Point)
%
% This function performs single-point crossover between pairs of
% individuals and returns the current generation after mating.
%
% Syntax:  NewChrom = xovsp(OldChrom, XOVR)
%
% Input parameters:
%    OldChrom   - Matrix containing the chromosomes of the old
%                 population. Each line corresponds to one individual
%                 (in any form, not necessarily real values).
```

```
%    XOVR        - Probability of recombination occurring between pairs
%                  of individuals.
%
% Output parameter:
%    NewChrom    - Matrix containing the chromosomes of the population
%                  after mating, ready to be mutated and/or evaluated,
%                  in the same format as OldChrom.

%  Author:     Hartmut Pohlheim
%  History:    28.03.94      file created

function NewChrom = xovsp(OldChrom, XOVR);

if nargin < 2, XOVR = NaN; end

% call low level function with appropriate parameters
   NewChrom = xovmp(OldChrom, XOVR, 1, 0);


% End of function
```

8.7 Algoritmo ERM_p

Nota: para usar este algoritmo se deberá incluir en la carpeta activa del Matlab la función ***crip.m***, la cual fue presentada anteriormente en la sección 8.6.2.1.

8.7.1 ermp.m

```
function [ResERMp] = ermp(P, Mat, NDesc, NInd);

%ermp returns a matrix containing the best models for the path of
lower relative standard deviation using
% the Enhanced Replacement Method, for all the individuals that form
the population.
%
%
%      Input:
%             P              Property vector
%             Mat            Descriptors matrix with descriptors pool
%             NDesc          Number of descriptors to use in the model
%             NInd           Number of individulas in the population
%
%     Returns:
%
%            ResERMp          Matrix containig the results of ERM for
every individual in the population.
%                             The first row shows the number of the
descriptor with lower der in the initial population.
%                             The second row show S of the optimal
model found, and the following rows show the
%                             descriptors in the model.
%
%
% Andrew G. Mercader
```

```
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2008



Inis=crtip(NInd, NDesc, Mat);


[c_r, r_r] = size(Inis);

ResERMp=[];
for k=1:c_r
    VecI=[];
    for i=1:r_r
    VecI=[VecI, Inis(k,i)];
    end
    [VecTot] = erm1(P, VecI, Mat);
    ResERMp=[ResERMp;VecTot];
end
ResERMp=sortrows(ResERMp,2);
%End
```

8.7.2 ermi.m

```
function [VecTot, TOT, time] = erm1(P, Vec, Mat)

%erm returns a matrix containing the best models for the path with
lower relative standard deviation using
% the Enhanced Replacement Method.
%TOT contains all the relative results of each step showing the
evolution of the method.
%
%
%      Input:
%             P             Property vector
%             Vec           Initial descriptors vector
%             Mat           Descriptors matrix with descriptors pool
%
%
%     Returns:
%
%            VecTot           vector containing the best model for all
the
%                              paths of the Replacement Method
%            TOT              contains all the relative results
%                              showing the evolution of the method.
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2007



TOT=[];
VecTot=[];
```

```
time=cputime;
warning off
[k_v,n_v]=size(Vec);


COEF=rmder(P,Vec,Mat);
COER=COEF;
pos=find(COEF==max(COER));

k=pos;

Sr=rms(P,Vec,Mat);
TOT(1).A(1,:)=[Sr,Vec];


VecA=rmsr(P, Vec, Mat, k);
Po(1)=k;
VecI=VecA;
if n_v==1
    VecTot=[1,VecI];
    TOT=VecI;
    time=cputime-time
    return
end
VecI(1)=[];
COEF=rmder(P,VecI,Mat);
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(2)=pos;
TOT(1).A(2,:)=VecA;
for i=2:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(1).A(i+1,:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end

for j=1:3;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(1).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
```

```
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
end
end

for j=4:100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(1).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(1).A(i+(j*n_v),:)==TOT(1).A(i+(j*n_v)-(2*n_v),:)
      break
   end

end

jj=j;
for j=jj:jj+100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rma2(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(1).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(1).A(i+(j*n_v),:)==TOT(1).A(i+(j*n_v)-(4*n_v),:)
        Po=[];
        break
   end
end
```

```
VecQ=find(TOT(1).A==min(TOT(1).A(:,1)));
VecI=[TOT(1).A(VecQ(1),:)];
VecI(1)=[];
jjj=j;
for j=jjj:jjj+100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==max(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(1).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==max(COER));
    Po(i+1)=pos;
    end
   if TOT(1).A(i+(j*n_v),:)==TOT(1).A(i+(j*n_v)-(2*n_v),:)
        Po=[];
        break
   end
end
VecP=find(TOT(1).A==min(TOT(1).A(:,1)));
VecTot(1,:)=[k,TOT(1).A(VecP(1),:)];
VecTot=sortrows(VecTot,2);

time=cputime-time
warning on

% end
```

8.8 Algoritmo ERM con solución de máximo S

8.8.1 ierm.m

```
function [VecTot, TOT, timei] = ierm(P, Mat, NumDesc)

%ierm returns a matrix containing the best models for all the paths of
the
%Enhanced Replacement Method using an initial set of descriptors with
a
%very high S, given by an inverse RM.
%TOT contains all the relative results of each step showing the
evolution of the method.
%
%
%       Input:
%               P               Property vector
%               Mat             Descriptors matrix with descriptors pool
```

```
%              NumDesc       Number of descriptors that the model will
have
%
%     Returns:
%
%             VecTot           vector containing the best model for all
the
%                                paths of the Replacement Method
%             TOT              contains all the relative results
%                               showing the evolution of the method.
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 12 Nov 2007


timei=cputime;

if (nargin < 3)
   error('the function requires at least 5 input variables. Type
''help ierm''.');
end




[c_m, r_m] = size(Mat);


NTot=r_m;

VecI=1:NumDesc;
lindep = ld(P, VecI,Mat);

if lindep==100
    for j=1:10000000000000000000000000000000000000
     VecI = randint(1,NumDesc,[1, NTot]);
     lindep = ld(P, VecI,Mat);
        if lindep==0
            break;
        end
    end
end

[VecT] = rmt_inv(P, VecI, Mat);
VecJ=VecT(NumDesc,3:end);

[VecTot, TOT, time] = erm(P, VecJ, Mat);


timei=cputime-timei
end

%End
```

8.8.2 Sub-funciones

8.8.2.1 rmsr_inv.m

```
function [Res] = rmsr_inv(P, VecI, Mat, Pth)

% rmsr_inv replaces all descriptors from Mat (except the ones in the
given descriptor vector) in the given position Pth and returns
%       the vector with higher S_err, using multiple linear regression
analysis.
%
%
%       Input:
%               P               Property vector
%               VecI            Descriptor vector
%               Mat             Descriptor matrix
%               Pth             Path to follow
%
%      Returns:
%
%               Res             Vector containing in the first place the
%                               S_err, and afterwards the corresponding
descriptor vector
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007


if (nargin < 4)
   error('rmsr_inv requires at least 4 input variables. Type ''help
rmsr_inv''.');
end

[k, n_m] = size(Mat);

Num=[1:n_m];
Vec2=VecI;
Vec2(Pth)=[];
Num(Vec2)=[];


[k,n_n]=size(Num);

Smax=0.00000000000000000000000000000001;   % A very big number
compared to a normal S necessary just to start the program
for j=1:n_n;
VecI(Pth)=Num(j);
Ser=rms_inv(P,VecI,Mat);
    if (Ser>Smax)
         Smax=Ser;
        desc=Num(j);
    end
end
VecI(Pth)=desc;
Res=[Smax, VecI];
```

```
%End of rmsr_inv
```

8.8.2.2 rmder_inv.m

```
function [COEF] = rmder_inv(P, Vec, Mat)

% rmder_inv returns a vector containing the relative standard
deviation of the regression coefficients. The
%                              constant of the regression is not
included
%
%
%       Input:
%              P              Property vector
%              Vec            Descriptors vector
%              Mat            Descriptors matrix
%
%
%      Returns:
%
%             COEF            Vector containing the standard deviation
for
%                              the regression coefficients without the
constant
%
%
%
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007



if (nargin < 3)
   error('rmder requires at least 3 input variables. Type ''help
rmder''.');
end


% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
% Check that independent (X) and dependent (P) data have compatible
dimensions
% ------------------------------------------------------------------
--------
     [n_x, k] = size(X);
[n_y,columns] = size(P);

if n_x ~= n_y,
    error('The number of rows in P must equal the number of rows in
Mat.');
end
```

```
if columns ~= 1,
    error('P must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
%  ------------------------------------------------------------------
     X = [ ones(n,1) X ]    ;


XTXI = X' * X;
if rcond(XTXI) < 1e-25
    S_err = 0.00000000000000000000000000000001;
    return;
end

 XTXI = inv(XTXI);
 Coefficients = XTXI * X' * P ;


%  Calculate the fitted regression values
%  --------------------------------------

    P_hat = X * Coefficients;



%  Calculate residuals and standard error
%  --------------------------------------

    residuals = P - P_hat;

           S_err = sqrt(residuals' * residuals / (n - k - 1) );

%  Calculate the standard deviation  for the regression coefficients
%  ------------------------------------------------------------------
----------

    covariance = XTXI .* S_err^2;

    C = sqrt(diag(covariance, 0));
    Er = abs(C./Coefficients)*100;



COEF=[[Er]'];
COEF(1)=[];
```

8.8.2.3 rmt_inv.m

```
function [VecTot, TOT, time] = rmt_inv(P, Vec, Mat)
%rmt_inv returns a matrix containing the best models for all the paths
of the Replacement Method used to maximize S (Inverse RM).
```

```
%TOT contains all the relative results showing the evolution of the
method.
%
%
%       Input:
%               P         Property vector
%               Vec       Initial descriptors vector
%               Mat       Descriptors matrix with descriptors pool
%
%
%      Returns:
%
%             VecTot           vector containing the best model for all
the
%                               paths of the Replacement Method
%             TOT              contains all the relative results
%                               showing the evolution of the method.
%
% Andrew G. Mercader, Pablo R. Duchowicz
% INIFTA, La Plata, Argentina
% Created: 5 March 2007


TOT=[];
VecTot=[];
time=cputime;
warning off

[k_v,n_v]=size(Vec);

for k=1:n_v

Sr=rms_inv(P,Vec,Mat);
TOT(k).A(1,:)=[Sr,Vec];

VecA=rmsr_inv(P, Vec, Mat, k);
Po(1)=k;
VecI=VecA;
if n_v==1
    VecTot=[1,VecI];
    TOT=VecI;
    time=cputime-time
    return
end
VecI(1)=[];
COEF=rmder_inv(P,VecI,Mat);
COER=COEF;
COER(Po)=[];
pos=find(COEF==min(COER));
Po(2)=pos;
TOT(k).A(2,:)=VecA;
for i=2:n_v;
    VecA=rmsr_inv(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+1,:)=VecA;
    VecI(1)=[];
    COEF=rmder_inv(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
```

```
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==min(COER));
    Po(i+1)=pos;
end

for j=1:2;
COER=COEF;
COER(Po)=[];
pos=find(COEF==min(COER));
Po(1)=pos;
for i=1:n_v;
    VecA=rmsr_inv(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder_inv(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==min(COER));
    Po(i+1)=pos;
end
end

for j=3:100;
COER=COEF;
COER(Po)=[];
pos=find(COEF==min(COER));
Po(1)=pos;
    for i=1:n_v;
    VecA=rmsr_inv(P, VecI, Mat, pos);
    VecI=VecA;
    TOT(k).A(i+(j*n_v),:)=VecA;
    VecI(1)=[];
    COEF=rmder_inv(P,VecI,Mat);
    if i==n_v
        Po=[];
        break
    end
    COER=COEF;
    COER(Po)=[];
    pos=find(COEF==min(COER));
    pos=pos(1);
    Po(i+1)=pos;
    end
   if TOT(k).A(i+(j*n_v),:)==TOT(k).A(i+(j*n_v)-(2*n_v),:)
        Po=[];
        break
   end

end
VecP=find(TOT(k).A==max(TOT(k).A(:,1)));
VecTot(k,:)=[k,TOT(k).A(VecP(1),:)];
VecTot=sortrows(VecTot,2);
end
time=cputime-time
```

```
warning on

% % End of RM
```

8.8.2.4 rms_inv.m

```
function  [S_err ] = rms_inv(P, Vec, Mat)

% rms_inv evaluates selection of descriptors using multiple linear
regression analysis.
%
%
%      Input:
%             P              Property vector
%             Vec            Descriptors vector
%             Mat            Descriptors matrix
%
%     Returns:
%
%             S_err            standard error
%
%
%
% Andrew G. Mercader
% INIFTA, La Plata, Argentina
% Created: 30 Jan 2007


if (nargin < 3)
   error('rms_inv  requires at least 3 input variables. Type ''help
rms_inv''.');
end


% Exctracts from descriptor matrix the matrix corresponding to the
given
% set of descriptors
% ------------------------------------------------------------------
--------
    X = Mat(:,Vec);
% Check that independent (X) and dependent (P) data have compatible
dimensions
% ------------------------------------------------------------------
--------
     [n_x, k] = size(X);
[n_y,columns] = size(P);

if n_x ~= n_y,
    error('The number of rows in P must equal the number of rows in
Mat.');
end

if columns ~= 1,
    error('P must be a vector, not a matrix');
end

n = n_x;

%  Solve for the regression coefficients using ordinary least-squares
```

```
%  ------------------------------------------------------------------------
      X = [ ones(n,1) X ]    ;

  XTXI = X' * X;
if rcond(XTXI) < 1e-18
    S_err = 0.0000000000000000000000000000000000000000000000001;
    return;
end

 XTXI = inv(XTXI);
       Coefficients = XTXI * X' * P ;


%  Calculate the fitted regression values
%  --------------------------------------

    P_hat = X * Coefficients;



%  Calculate residuals and standard error
%  --------------------------------------

    residuals = P - P_hat;

     S_err = sqrt(residuals' * residuals / (n - k - 1) );
```

Referencias

1. M. Head-Gordon, E. Artacho, *Chemistry on the computer.* Physics Today, **2008**, 61, 4: 58-63.
2. E. Clementi, *Computational Aspects for Large Chemical Systems*. Lecture Notes in Chemistry Vol.19. **1980**, New York, Springer-Verlag.
3. C. Hansch, A. Leo, *Exploring QSAR: Fundamentals and Applications in Chemistry and Biology*. **1995**, Washington, D.C., Am. Chem. Soc.
4. L. B. Kier, *Molecular Orbital Theory in Drug Research*. **1971**, New York, Academic Press.
5. D. Bonchev, *Information Theoretic Indices for Characterization of Chemical Structures*, ed. R.S. Press. **1983**, Chichester, Wiley.
6. D. Bonchev, *Graph Theoretical Approaches to Chemical Reactivity*, ed. O. Mekenyan. **1994**, Dordrecht, Kluwer Academic Publishers.
7. A. Crum-Brown, T. R. Fraser, Trans. R. Soc. Edinburgh, **1868**, 25: 151.
8. C. Richet, C. R. Seancs, Soc. Biol. Ses. Fil, **1893**, 9: 775.
9. H. Meyer, Arch. Exp. Pathol. Pharmakol., **1899**, 42: 109.
10. E. Overton, *Studien Uber die Narkose*. **1901**, Germany, Fischer Jena.
11. J. Ferguson, Proc. R. Soc. London Ser. B, **1939**, 127: 387.
12. P. H. Bell, J. R. O. Roblin, J. Am. Chem. Soc., **1942**, 64: 2905.
13. A. Albert, S. Rubbo, R. Goldacre, M. Darcy, J. Stove, Br. J. Exp. Pathol., **1945**, 26: 160.
14. A. Albert, *Selective Toxicity: The Physicochemical Bases of Therapy, 7th ed.* **1985**, London, Chapman and Hall.
15. L. P. Hammett, Chem. Rev., **1935**, 17: 125.
16. L. P. Hammett, *Physical Organic Chemistry 2nd ed.* **1970**, New York, McGraw-Hill.
17. R. W. Taft, J. Am. Chem. Soc., **1952**, 74: 3120.
18. C. Hansch, T. Fujita, *ρ-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure.* J. Am. Chem. Soc, **1964**, 86: 1616-1626.
19. W. Karcher, J. Devillers, *Practical Applications of Quantitative Structure-Activity Relationships (QSAR) in Environmental Chemistry and Toxicology*. **1990**, Dordrecht, Kluwer Academic Publications.
20. R. Carbó-Dorca, D. Robert, L. I. Amat, X. Jirones, E. Besalú, *Molecular Quantum Similarity in QSAR and Drug Research*. Springer-Verlag. **2000**, Berlin.
21. C. M. Auer, J. B. Nabholz, K. P. Baetcke, Environ. Health Perspect., **1990**, 87: 183.
22. S. C. Basak, G. D. Grunwals, B. D. Gute, K. Balasubramanian, D. J. Ortiz, Chem. Inf. Comput. Sci., **2000**, 40: 885.
23. J. K. Seydel, *QSAR and Strategies in the Design of Bioactive Compounds*. **1985**, Weinheim, VCH.
24. J. G. Topliss, *Quantitative Structure-Activity Relationships of Drugs*. **1983**, New York,, Academic Press.
25. E. R. Malinowski, *Factor Analysis in Chemistry*. **1991**, New York, Wiley.
26. H. Hotelling, J. Educ. Psychol, **1933**, 24: 417.

27. S. Wold, M. Sjostrom, L. Eriksson, *Encyclopedia of Computational Chemistry*. **1998**, Chichester, England, Wiley.
28. H. V. d. Waterbeemd, *Chemometric Methods in Molecular Design*. **1995**, Weinheim, VCH.
29. H. Kubiny, *QSAR: Hansch Analysis and Related Approaches*. **1993**, Weinheim, VCH.
30. H. V. d. Waterbeemd, *Advanced Computer- Assisted Techniques in Drug Discovery*. **1995**, Weinheim, VCH.
31. M. Randic, J. Math. Chem., **1991**, 7: 155.
32. R. Todeschini, V. Consonni, *Handbook of Molecular Descriptors*. **2000**, Weinheim, Germany, Wiley VCH.
33. D. Rogers, A. J. Hopfinger, J. Chem. Inf. Comput. Sci., **1994**, 34: 854.
34. A. Hoskuldsson, Chemom. Intell. Lab. Syst., **1996**, 32: 37.
35. R. Hoffmann, V. I. Minkin, B. K. Carpenter, Bull. Soc. Chim. Fr., **1996**, 133: 117.
36. D. M. Hawkins, S. C. Basak, D. Mills, J. Chem. Inf. Model., **2003**, 43: 579-586.
37. L. B. Kier, L. H. Hall, *Molecular Connectivity in Chemistry and Drug Research*. **1976**, New York, Academic Press.
38. N. Trinajstic, S. Nikolic, B. Lucic, D. Amic, Acta Pharm., **1996**, 46: 249.
39. A. Sabljic. In Topological indices and environmental chemistry. **1990**, Dordrecht, Kluwer Academic Publ.
40. Z. Mihalic, N. Trinajstic, J. Chem. Educ., **1992**, 69: 701.
41. T. Engel, *Representation of Chemical Compounds*, in *Chemoinformatics*, J. Gasteiger and T. Engel, Editors. **2003**, Wiley-VCH: Wenheim.
42. A. Beck, M. N. Bleicher, D. W. Crowe, *Excursions Into Mathematics: The Millennium Edition* **2000**, Wellesley, Massachusetts, A K Peters, Ltd.
43. L. Euler, *Solutio Problematis ad Geometriam Situs Pertinentis*. Commentarii Academiae Scientiarum Imperialis Petropolitanae, **1736**, 8: 128-140.
44. A. T. Balaban, *Chemical Graphs: Looking Back and Glimpsing Ahead*. J. Chem. Inf. Comput. Sci., **1995**, 35, 3: 339-350.
45. O. Ivanciuc, *Coding the constitution: Graph Theory in chemistry*, in *Handbook of Chemoinformatics*, J. Gasteiger, Editor. **2003**, Wiley-VCH: Weinheim.
46. H. P. Schultz, *Topological organic chemistry. 1. Graph theory and topological indices of alkanes*. J. Chem. Inf. Comput. Sci., **1989**, 29, 3: 227-228.
47. J. Dugundji, I. Ugi, *An algebraic model of constitutional chemistry as a basis for chemical computer programs*, in *Computers in Chemistry*. **1973**, 19-64.
48. C. Morley, *OpenBabel 2.2.0* http://openbabel.org/wiki/Main_Page. **2006**.
49. HYPERCHEM, 6.03 (Hypercube) http://www.hyper.com.
50. V. N. Viswanadhan, A. K. Ghose, G. R. Revankar, R. K. Robins, *Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics*. J. Chem. Inf. Comput. Sci., **1989**, 29, 3: 163-172.
51. G. B. Moreau, P., Nouv. J. Chim., **1980**, 4: 359.
52. G. Moreau, P. Broto, Nouv. J. Chim., **1980**, 4: 757-764.
53. R. B. Frank, *A Chemically Intuitive Molecular Index Based on the Eigenvalues of a Modified Adjacency Matrix*. Quantitative Structure-Activity Relationships, **1997**, 16, 4: 309-314.

54. F. R. Burden, *Molecular identification number for substructure searches.* J. Chem. Inf. Comput. Sci., **1989**, 29, 3: 225-227.
55. R. S. Pearlman, K. M. Smith, *Metric Validation and the Receptor-Relevant Subspace Concept.* J. Chem. Inf. Comput. Sci., **1999**, 39, 1: 28-35.
56. J. Galvez, R. Garcia, M. T. Salavert, R. Soler, J. Chem. Inf. Comput. Sci., **1994**, 34: 520-525.
57. J. Galvez, R. Garcia-Domenech, C. de Gregorio Alapont, V. De Julian-Ortiz, L. Popa, J. Mol. Graphics, **1996**, 14: 272-276.
58. J. Galvez, R. Garcia-Domenech, V. De Julian-Ortiz, R. Soler, J. Chem. Inf. Comput. Sci., **1995**, 35: 272-284.
59. I. Rios-Santamarina, R. Garcia-Domenech, J. Galvez, J. Cortijo, P. Santamaria, E. Marcillo, Bioorg. Med. Chem. Lett., **19988**: 477-482.
60. L. B. Kier, L. H. Hall, *Molecular Connectivity in Structure-Activity Analysis.* **1986**, New York, Wiley.
61. A. Radecki, H. Lamparczyk, R. Kaliszan, Chromatographia, **1979**, 12: 597.
62. M. Randic, New J. Chem., **1995**, 19: 781.
63. V. Consonni, R. Todeschini, M. Pavan, J. Chem. Inf. Model., **2002**, 42: 693.
64. R. Todeschini, P. Gramatica, *SD-modelling and Prediction by WHIM Descriptors. Part 5. Theory Development and Chemical Meaning of WHIM Descriptors.* Quantitative Structure-Activity Relationships, **1997**, 16, 2: 113-119.
65. P. Gramatica, M. Corradi, V. Consonni, Chemosphere, **2000**, 41: 763-777.
66. J. Gasteiger, J. Sadowski, J. Schuur, P. Selzer, L. Steinhauer, V. Steinhauer, *Chemical Information in 3D Space.* J. Chem. Inf. Comput. Sci., **1996**, 36, 5: 1030-1037.
67. J. H. Schuur, P. Selzer, J. Gasteiger, *The Coding of the Three-Dimensional Structure of Molecules by Molecular Transforms and Its Application to Structure-Spectra Correlations and Studies of Biological Activity.* J. Chem. Inf. Comput. Sci., **1996**, 36, 2: 334-344.
68. A. R. Katritzky, L. Mu, V. S. Lobanov, *Correlation of Boiling Points with Molecular Structure. 1. A Training Set of 298 Diverse Organics and a Test Set of 9 Simple Inorganics.* J. Phys. Chem., **1996**, 100: 10400-10407.
69. O. Mekenyan, D. Peitchev, D. Bonchev, N. Trinajstic, I. P. Bangov, Arzneim.Forsch, **1986**, 36: 176-183.
70. J. Kruszewski, T. M. Krygowski, Tetrahedron Lett., **1972**, 36: 3839-3842.
71. M. Karelson, V. S. Lobanov, A. R. Katritzky, Chem.Rev., **1996**, 96: 1027-1043.
72. Excel. **2007**, Microsoft http://office.microsoft.com/es-es/excel/.
73. P. R. Duchowicz, A. G. Mercader, F. M. Fernández, E. A. Castro, *Prediction of aqueous toxicity for heterogeneous phenol derivatives by QSAR* Chemometrics and Intelligent Laboratory Systems **2008**, 90: 97–107.
74. N. R. Draper, H. Smith, *Applied Regression Analysis.* **1981**, New York, John Wiley&Sons.
75. S. S. So, M. Karplus, *Evolutionary Optimization in Quantitative Structure-Activity Relationship: An Application of Genetic Neural Networks* J. Med. Chem., **1996**, 39: 1521-1530.
76. M. Melanie, *An Introduction to Genetic Algorithms.* **1998**, A Bradford Book The MIT Press: Cambridge, Massachusetts • London, England, 3-9,130 -131.
77. P. R. Duchowicz, E. A. Castro, F. M. Fernández, *Alternative Algorithm for the Search of an Optimal Set of Descriptors in QSAR-QSPR Studies.* MATCH Commun. Math. Comput. Chem., **2006**, 55: 179-192.

78. P. R. Duchowicz, M. Fernández, J. Caballero, E. A. Castro, F. M. Fernández, *QSAR of Non-Nucleoside Inhibitors of HIV-1 Reverse Transcriptase.* Bioorg. Med. Chem., **2006**, 14: 5876-5889.
79. A. M. Helguera, P. R. Duchowicz, M. A. C. Pérez, E. A. Castro, M. N. D. S. Cordeiro, M. P. González, *Application of the Replacement Method as Novel Variable Selection Strategy in QSAR. 1. Carcinogenic Potential.* Chemometr. Intell. Lab., **2006**, 81: 180-187.
80. A. G. Mercader, P. R. Duchowicz, M. A. Sanservino, F. M. Fernandez, E. A. Castro, *QSPR analysis of fluorophilicity for organic compounds.* Journal of Fluorine Chemistry, **2007**, 128, 5: 484-492.
81. P. R. Duchowicz, A. G. Mercader, F. M. Fernández, E. A. Castro, *Prediction of aqueous toxicity for heterogeneous phenol derivatives by QSAR* Chemom. Intell. Lab. Syst, **2007**, 90: 97–107.
82. A. G. Mercader, P. R. Duchowicz, F. M. Fernandez, E. A. Castro, *Modified and enhanced replacement method for the selection of molecular descriptors in QSAR and QSPR theories.* Chemom. Intell. Lab. Syst. , **2008**, 92: 138-144.
83. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *Equation of State Calculations by Fast Computing Machines.* J. Chem. Phys., **1953**, 21, 6: 1087-1092.
84. S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, *Optimization by Simulated Annealing.* Science, **1983**, 220, 4598: 671-680.
85. P. R. Duchowicz, M. G. Vitale, E. A. Castro, J. C. Autino, G. P. Romanelli, D. O. Bennardi, *QSAR Modeling of the Interaction of Flavonoids with GABA(A) Receptor.* Eur. J. Med. Chem., **2007**, 43, 8: 1593-1602
86. I. O. Edafiogho, C. N. Hinko, H. Chang, J. A. Moore, D. Mulzac, J. M. Nicholson, K. R. Scott, *Synthesis and anticonvulsant activity of enaminones.* J. Med. Chem., **1992**, 35, 15: 2798-2805.
87. I. O. Edafiogho, K. V.V, Ananthalakshmi, S. B. Kombian, *Anticonvulsant evaluation and mechanism of action of benzylamino enaminones* Bioorganic & Medicinal Chemistry, **2006** 14, 15: 5266-5272.
88. N. D. Eddington, D. S. Cox, M. Khurana, N. N. Salama, J. P. Stables, S. J. Harrison, A. Negussie, R. S. Taylor, U. Q. Tran, J. A. Moore, J. C. Barrow, K. R. Scott, *Synthesis and anticonvulsant activity of enaminones Part 7. Synthesis and anticonvulsant evaluation of ethyl 4-[(substituted phenyl)amino]-6-methyl-2-oxocyclohex-3-ene-1-carboxylates and their corresponding 5-methylcyclohex-2-enone derivatives* European Journal of Medicinal Chemistry, **2003**, 38, 1: 49-64.
89. L. Jäntschi, *QSPR on Estimating of Polychlorinated Biphenyls Relative Response Factor using Molecular Descriptors Family.* Leonardo Electronic Journal of Practices and Technologies, **2007**, 3, 5: 67 - 84.
90. A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca, eds. *Genetic Algorithm TOOLBOX For Use with MATLAB User's Guide v1.2.* **1994**: Sheffield, http://www.shef.ac.uk/acse/research/ecrg/gat.html.
91. A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, D. O. Bennardi, J. C. Autino, G. P. Romanelli, *QSAR prediction of inhibition of aldose reductase for flavonoids.* Bioorganic & Medicinal Chemistry, **2008**, 16: 7470–7476
92. A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, E. Wolcan, *QSPR Study of solvent quenching of the $^5D_0 \rightarrow {}^7F_2$ emission of Eu(6,6,7,7,8,8,8-*

heptafluoro-2,2-dimethyl-3,5-octanedionate)$_3$. Chem. Phys. Lett., **2008**, 462: 352–357.
93. A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, F. M. Cabrerizo, A. H. Thomas, *Predictive Modeling of the Total Deactivation Rate Constant of Singlet Oxygen by Heterocyclic Compounds.* Journal of Molecular Graphics and Modelling, **2009**submitted.
94. D. M. Hawkins, *The Problem of Overfitting.* Journal of Chemical Information and Computer Sciences, **2004**, 44, 1: 1-12.
95. H. Kubinyi, *Variable Selection in QSAR Studies. I. An Evolutionary Algorithm.* QSAR & Combinatorial Science, **1994**, 13, 3: 285-294.
96. H. Kubinyi, *Variable Selection in QSAR Studies. II. A Highly Efficient Combination of Systematic Search and Evolution.* QSAR & Combinatorial Science, **1994**, 13, 4: 393-401.
97. A. G. Mercader, *Selection of an optimal set of descriptors: use of the Enhanced Replacement Method* in *QSPR-QSAR STUDIES ON DESIRED PROPOERTIES FOR DRUG DESIGN*, E.A. Castro, Editor. **2009**, Signpost Design: India.
98. C. Hansch, *Comprehensive Drug Design.* Vol. 6. **1990**, New York, Pergamon Press.
99. J. Topliss, R. Costello, J. Med. Chem., 15: 1066.
100. C. Mosier, Educ. Psychological Measurement **1951**, 11: 5.
101. D. M. Allen, Technometrics, **1968**, 16: 125.
102. S. Geisser, J. Am. Stat. Assoc., **1975**70: 320.
103. P. A. Lachenbruch, M. Mickey, Technometrics, **1968**, 10: 1.
104. A. Golbraikh, A. Tropsha, *Beware of q2!* Journal of Molecular Graphics and Modelling, **2002**, 20, 4: 269-276.
105. S. Wold, L. Eriksson, *Statistical validation of QSAR results*, in *Chemometrics Methods in Molecular Design*, H.v.d. Waterbeemd, Editor. **1995**, VCH: Weinheim, 309-318.
106. I. T. Horvath, J. Rabai, Science, **1994**, 72: 266.
107. D. S. Jozsef Rabai, E. K. Borbas, I. Kovesi, I. Kovesdi, A. G. Antal Csampai, V. E. Pashinnik, Y. G. Shermolovich, J. Fluorine Chem., **2002**, 114: 199-207.
108. C. D. R. Rocaboy, B. L. Bennett, J. A. Gladysz, J. Phys. Org. Chem., **2000**, 13: 596-603.
109. P. C. Bhattacharyya, B.; Fawcett, J.; Gudmunsen, D.; Hope, E. G.; Kemmitt, R. D. W.; Paige, D. R.; Russell, D. R.; Stuart, A. M.; Wood, D. R. W., J. Fluorine Chem., **2000**, 101: 247-255.
110. L. E. Kiss, Kövesdi, I., Rábai, J., J. Fluorine Chem., **2001**, 108: 95.
111. F. T. T. Huque, Jones, K., Saunders, R. A., Platts, J. A., J. Fluorine Chem., **2002**, 115: 119-128.
112. P. R. Duchowicz, Fernández, F. M., Castro, E. A., J. Fluorine Chem., **2004**, 125: 43-48.
113. E. de Wolf, Ruelle, P., van den Brocke, J., Deelman, B., van Koten, G., J. Phys. Chem., **2004**, 108: 1458.
114. M. S. Daniels, Saunders, R. A., Platts, J. A., J. Fluorine Chem., **2004**, 125: 1291-1298.
115. D. Szabó, J. Mohl, A. M. Bálint, A. Bodor, J. Rábai, J. Fluorine Chem., **2006**, 127: 1496–1504.
116. P. R. Duchowicz, E. A. Castro, F. M. Fernández, M. P. González, *A New Search Algorithm of QSPR/QSAR Theories: Normal Boiling Points of Some Organic Molecules.* Chem. Phys. Lett., **2005**, 412: 376-380.

117. H. Akaike, *Information Theory and and Extension of the Maximum Likelihood Principle*, in *Second International Symposium on Information Theory*, B.N. Petrov, Csáki, F., Editor. **1973**, Akademiai Kiado: Budapest, 267-281.
118. H. Akaike, IEEE Trans. Automat. Control, **1974**, AC-19: 716-723.
119. DRAGON, 5.0 Evaluation Version http://www.disat.unimib.it/chm.
120. R. Todeschini, Consonni, V., *Handbook of Molecular Descriptors*. Methods and Principles in Medicinal Chemistry, ed. R. Mannhold, Kubinyi, H., Timmerman, H. Vol. 11, Wiley VCH.
121. M. L. S. Hanson, K. R., Environ. Sci. Technol., **2002**, 36: 3257-3264.
122. S. F. Smith, V. J.; Layiwola, P. J.; Menezes-Filho, J. A. , Chemosphere, **1994**, 28: 825-836.
123. S. P. Bradbury, Toxicol. Lett., **1995**, 79: 229-237.
124. M. T. D. A. Cronin, A. O.; Duffy, J. C.; Netzeva, T. I.; Rowe, P. H.; Valkova, I. V.; Schultz, T. W. , Chemosphere, **2002**, 49: 1201-1221.
125. J. S. Damborsky, T. W. , Chemosphere, **1997**, 34: 429-446.
126. K. B. Pirselova, S.; Schultz, T. W. , Arch. Environ. Contam. Toxicol., **1996**, 30: 170-177.
127. J. Devillers, SAR&QSAR Environ. Res., **2004**, 15: 237-249.
128. Matlab, 5.0 The MathWorks Inc. http://www.mathworks.com/.
129. K. H. Gabbay, *The sorbitol pathway and the complications of diabetes.* N. Engl. J. Med., **1973**, 288, 16: 831-836.
130. J. H. Kinoshita, S. D. Varma, H. N. Fukui, *Aldose Reductase in Diabetes.* Jap. J. Ophtal., **1976**, 20: 399.
131. J. H. Kinoshita, *Mechanisms Initiating Cataract Formation Proctor Lecture.* Invest. Ophthalmol. Vis. Sci., **1974**, 13, 10: 713-724.
132. S. D. Varma, J. H. Kinoshita, *Inhibition of lens aldose reductase by flavonoids-Their possible role in the prevention of diabetic cataracts.* Biochemical Pharmacology **1976**, 25, 22: 2505-2513.
133. M. M. Iwu, O. A. Igboko, C. O. Okunji, M. S. Tempesta, *Antidiabetic and aldose reductase activities of Biflavanones of Garcinia kola.* Pharm. Pharmacol., **1990**, 42: 290-292.
134. J. Okuda, I. Miwa, K. Inagaki, T. Horie, M. Nakayama, *Inhibition of aldose reductase by 3',4'-dihydroxyflavones.* Chem. Pharm. Bull., **1984**, 32, 2: 767-772.
135. S. D. Varma, *Inhibition of Aldose Reductase by Flavonoids: Possible Attenuation of Diabetic Complications.* , in *Plant Flavonoids in Biology and Medicine: Biochemical, Pharmacological, and Structure–Activity Relationships*, A.R. Liss, Editor. **1986**: New York, 343-357.
136. A. Štefanič-Petek, A. Krbavčič, T. Šolmajer, *QSAR of Flavonoids: 4. Differential Inhibition of Aldose Reductase and $p56^{lck}$ Protein Tyrosine Kinase.* Croat. Chem. Acta, **2002**, 75, 2: 517-529.
137. M. Fernández, C. J., A. Morales Helguera, E. A. Castro, M. Pérez Gonzáez, *Quantitative structure–activity relationship to predict differential inhibition of aldose reductase by flavonoid compounds.* Bioorganic & Medicinal Chemistry, **2005**, 13: 3269-3277.
138. D. O. Bennardi, G. P. Romanelli, J. L. Jios, P. G. Vazquez, C. V. Caceres, J. C. Autino, *Synthesis of substituted flavones and aryl-chromones using p and si keggin heteropoly-acids as catalysts.* Heterocyclic Communications, **2007**, 13, 1: 79-82.

139. J. Okuda, I. Miwa, K. Inagaki, T. Horie, M. Nakayama, *Inhibition of Aldose Reductase from Rat and Bovine Lenses by Flavanoids.* Biochemical Pharmacology, **1982**, 31, 23: 3807-3822.
140. K. Inagaki, I. Miwa, J. Okuda, *Affinity purification and glucose specificity of aldose reductase from bovine lens* **1982**, 216, 1: 337-344.
141. P. F. Kador, L. O. Merola, J. H. Kinoshita, *Differences in the susceptibility of various aldose reductases to inhibition.* Docum. Ophthalmol Proc. Series, **1979**, 18: 117.
142. S. Sung Lim, S. Hoon Jung, J. Ji, K. H. Shin, S. R. Keum, *Synthesis of flavonoids and their effects on aldose reductase and sorbitol accumulation in streptozotocin-induced diabetic rat tissues.* Journal of Pharmacy and Pharmacology, **2001**, 53: 653-668.
143. H. Liu, P. Gramatica, *QSAR study of selective ligands for the thyroid hormone receptor β.* Bioorganic & Medicinal Chemistry, **2007**, 15, 15: 5251-5261.
144. M. J. Lochhead, P. R. Wamsley, K. L. Bray, *Luminescence Spectroscopy of Europium(III) Nitrate, Chloride, and Perchlorate in Mixed Ethanol-Water Solutions.* Inorg. Chem., **1994**, 33, 9: 2000-2003.
145. P. R. Selvin, J. E. Hearst *Luminescence energy transfer using a terbium chelate: improvements on fluorescence energy transfer.* Proc. Natl. Acad. Sci. USA **1994**, 91: 10024-10028.
146. P. R. Selvin, Annu. Rev. Biophys. Biomol. Struct., **2002**, 31: 275.
147. M. Li, P. R. Selvin, *Luminescent Polyaminocarboxylate Chelates of Terbium and Europium: The Effect of Chelate Structure.* J. Am. Chem. Soc., **1995**, 117, 31: 8132-8138.
148. I. Hemmila, V. Laitala, *Progress in Lanthanides as Luminescent Probes.* Journal of Fluorescence, **2005**, 15, 4: 529-542.
149. L. S. Villata, E. Wolcan, M. R. Féliz, A. L. Capparelli, J. Phys. Chem. A, **1999**, 103: 5661.
150. L. S. Villata, E. Wolcan, M. R. Féliz, A. L. Capparelli, *Solvent quenching of the $^5D_0 \rightarrow {}^7F_2$ emission of Eu(6,6,7,7,8,8,8-heptafluoro-2,2-dimethyl-3,5-octanedionate)$_3$* Journal of Photochemistry and Photobiology A: Chemistry **1998**, 115, 2: 185-189.
151. E. L. Clennana, A. Paceb, *Advances in singlet oxygen chemistry.* Tetrahedron, **2005**, 61: 6665–6691.
152. C. Schweitzer, R. Schmidt, *Physical Mechanisms of Generation and Deactivation of Singlet Oxygen.* Chem. Rev., **2003**, 103: 1685-1757.
153. C. S. Foote, E. L. Clennan, *Properties and reactions of singlet dioxygen, in Active Oxygen in Chemistry*, ed. C.S. Foote, et al. Vol. 2, Ch. 4. **1995**, New York, Chalman & Hall.
154. M. C. DeRosa, R. J. Crutchley, *Photosensitized singlet oxygen and its applications.* Coor. Chem. Rev., **2002**, 233–234: 351–371.
155. K. Briviba, L. O. Klotz, H. Sies, *Toxic and signaling effects of photochemically or chemically generated singlet oxygen in biological systems.* Biol. Chem., **1997**, 378: 1259-1265.
156. E. Cadenas, *Biochemistry of oxygen toxicity.* Annu. Rev. Biochem., **1989**, 58: 79-110.
157. A. U. Kahn, *Direct spectroscopic observation of 1.27 mm and 1.58 mm emission of singlet ($^1\Delta_g$) molecular oxygen in chemically generated and dye-photosensitized liquid solutions at room temperature.* Chem. Phys. Lett., **1980**, 72: 112–114.

158. A. H. Thomas, C. Lorente, A. L. Capparelli, C. G. Martínez, A. M. Braun, E. Oliveros, *Singlet oxygen ($^1\Delta_g$) production by pterin derivatives in aqueous solutions.* Photochem. Photobiol. Sci, **2003**, 2: 245 - 250.
159. C. Lorente, A. H. Thomas, *Photophysics ans Photochemistry of Pterins in Aqueous Solution.* Acc. Chem. Res, **2006**, 39: 395-402.
160. F. M. Cabrerizo, M. L. Dántola, G. Petroselli, A. L. Capparelli, A. H. Thomas, A. M. Braun, C. Lorente, E. Oliveros, *Reactivity of Conjugated and Unconjugated Pterins with Singlet Oxygen ($O_2(^1\Delta_g)$): Physical Quenching and Chemical Reaction.* Photochem. Photobiol. Sci, **2007**, 83: 526–534.
161. M. L. Dantola, A. H. Thomas, A. M. Braun, E. Oliveros, C. Lorente, *Singlet Oxygen ($O_2(^1\Delta_g)$) Quenching by Dihydropterins.* J. Phys. Chem. A, **2007**, 111, 20: 4280-4288.
162. F. Wilkinson, P. W. Helman, A. B. Ross, *Rate Constants for the Decay and Reactions of the Lowest Electronically Excited Singlet State of Molecular Oxygen in Solution. An Expanded and Revised Compilaiton.* Journal of Physical and Chemical Reference Data, **1995**, 24, 2: 663-677.
163. e-Dragon, Electronic remote version of Dragon 5.4, <http://www.vcclab.org/lab/edragon/>.
164. K. C. Pari, S. Sundari, S. Chandani, D. Balasubramanian, *β-Carbolines That Accumulate in Human Tissues May Serve a Protective Role against Oxidative Stress**. The Journal of Biological Chemistry, **2000**, 275, 4: 2455–2462.
165. A. R. Leach, V. J. Gillet, *An Introduction to Chemoinformatics*. **2007**, Dordrecht, The Netherlands., Springer.
166. Derive. **2000**, 5, Texas Instrument Incorporated, http://www.derive-europe.com/specs.asp?d6.
167. J. García de Jalón, J. Ignacio Rodríguez, R. Goñi Lasheras, A. Brazález Gerra, P. Funes Martínez, R. Rodríguez Tamayo, *Aprenda Lenguaje ansi C como si estuviera en primero*, U.d.N. Escuela Superior de Ingenieros Industriales, Editor. **1998**: San Sebastián, http://mat21.etsii.upm.es/ayudainf/aprendainf/AnsiC/leng_c.pdf.
168. J. García de Jalón, J. Ignacio Rodríguez, *Aprenda Matlab 7.0 como si estuviera en primero*, U.P.d.M. Escuela Técnica Superior de Ingenieros Industriales, Editor. **2005**: Madrid, http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf.

Publicaciones

El presente trabajo de tesis dio a lugar a las publicaciones detalladas a continuación:

- "Selection of an optimal set of descriptors: use of the Enhanced Replacement Method", A. G. Mercader en "QSPR-QSAR STUDIES ON DESIRED PROPOERTIES FOR DRUG DESIGN", Signpost Design, India, 2009, E. A. Castro Editor-in-Chief.
- "Predictive Modeling of the Total Deactivation Rate Constant of Singlet Oxygen by Heterocyclic Compounds", A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, F.M. Cabrerizo, A. H. Thomas, Journal of Molecular Graphics and Modelling, 2009. (submitted)
- "QSPR Study of solvent quenching of the $^5D_0 \rightarrow {}^7F_2$ emission of Eu(6,6,7,7,8,8,8-heptafluoro-2,2-dimethyl-3,5-octanedionate)$_3$", A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, E. Wolcan, Chemical Physics Letters, 462 (2008) 352–357.
- "QSAR Prediction of inhibition of aldose reductase for flavonoids", A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, D. O. Bennardi, J. C. Autino, G. P. Romanelli, Bioorganic & Medicinal Chemistry, 16 (2008) 7470-7476.
- "Modified and Enhanced Replacement Method for the selection of Molecular Descriptors in QSAR and QSPR theories", A. G. Mercader, P. R. Duchowicz, F. M. Fernández, E. A. Castro, Chemometrics and Intelligent Laboratory Systems, 92 (2008) 138-144, Elsevier
- "Prediction of Aqueous Toxicity for Heterogeneous Phenol Derivatives by QSAR", P. R. Duchowicz, A. G. Mercader, F. M. Fernández, E. A. Castro, Chemometrics and Intelligent Laboratory Systems, 90 (2008) 97–107, Elsevier
- "QSPR Analysis of Fluorophilicity for Organic Compounds", A. G. Mercader, P. R. Duchowicz, M. A. Sanservino, F. M. Fernandez, E. A. Castro, Journal of Fluorine Chemistry, 128 (2007) 484-492, Elsevier.
- "Calculation of total electronic energies from correlation weighting of local graph invariants", A. G. Mercader, E. A. Castro, A. A. Toropov, Journal of Molecular Modeling (2001), 7, 1-5, Springer-Verlag, Online.
- "Maximum Topological Distances Based Indices as Molecular Descriptors for QSPR. 4. Modeling the Enthalpy of Formation of Hydrocarbons from Elements", A. G. Mercader, E. A. Castro, A. A. Toropov, International Journal of Molecular Sciences, (2001), 2, 121-132, MDPI, Basel, Switzerland.
- "Improved Correlations Between 19F-NMR Chemical Shifts and Physical Chemistry Properties in Flouorohalohydrocarbons", A. G. Mercader, E. A. Castro, Revue Roumaine de Chimie, (2001), 46, 12, 1285-1292,Editura Academiei Romane, Bucarest.
- "QSPR modeling of the enthalpy of formation from elements by means of correlation weighting of local invariants of atomic orbital molecular graphs", A. G. Mercader, E. A. Castro, A. A. Toropov, Chemical Physics Letters, (2000), 330, 612-623, ELSEVIER, Amsterdam

Asimismo se han presentado los siguientes trabajos en congresos nacionales e internacionales:

- 3er Workshop Argentino de Química Medicinal, Asociación Química Argentina, Los Cocos, Córdoba, 10 y 11 de Noviembre de 2008. Presentación del Poster: "Estudio QSAR sobre la lipofilicidad de di-N-óxidos de quinoxalinas en el tratamiento de tuberculosis", Duchowicz P. R., Vicente E., Mercader A. G., Pérez-Silanes S., Aldana I., Fernández F. M., Castro E. A., Monge A.
- XXVII Congreso Argentino de Química, Universidad Nacional de Tucumán, Tucumán, 17-19 de Septiembre de 2008. Presentación del Poster: "Estudio QSPR del efecto del solvente en la desactivación de la emisión $^5D_0 \rightarrow {}^7F_2$ del Eu(6,6,7,7,8,8,8-heptafluoro-2,2-dimetil-3,5-octanodionato)$_3$", Andrew G Mercader, Pablo R. Duchowicz, Francisco M. Fernández, Eduardo A. Castro y Ezequiel Wolcan
- V Congreso Iberoamericano de Física y Química Ambiental, Sociedad Iberoamericana de Física y Química Ambiental (SiFyQA), Mar del Plata 14-18 de Abril de 2008. Defensa del Poster: "Estudio QSPR del Factor de Respuesta Relativo en Bifenilos Policlorinados", Andrew G. Mercader, Pablo R. Duchowicz, Francisco M. Fernández y Eduardo A. Castro
- Semana de divulgación del conocimiento, Facultad de ciencias exactas, UNLP, La Plata 10-14 Diciembre de 2007, Exposición del trabajo: "Introducción y avances en las teorías QSAR/QSPR", Andrew G. Mercader, Pablo R. Duchowicz, Francisco M. Fernández y Eduardo A. Castro.
- NSF Pan-American Advanced Studies Institute (PASI) on Sustainability and Green Chemistry Summer School, ACS, México D.F. 29 Mayo-10 Junio de 2007. Defensa del poster: "Introduction and Progress in linear QSAR/QSPR Modeling", Andrew G. Mercader, Pablo R. Duchowicz, Francisco M. Fernández y Eduardo A. Castro
- XV Congreso Argentino de Fisicoquímica y Química Inorgánica, Asociación Argentina de Investigación Fisicoquímica, Tandil 17-20 de Abril de 2007. Defensa del Poster: "Estudio QSPR de Constantes de Desactivación Física y Química de Oxígeno Singlete por compuestos heterocíclicos", Andrew G. Mercader, Franco M. Cabrerizo, Pablo R. Duchowicz, Andrés H. Thomas, Eduardo A. Castro y Francisco M. Fernández
- XXVI Congreso Argentino de Química, Asociación Química Argentina, San Luis, 13-15 de Septiembre de 2006. Defensa del Poster: "Predicción de Toxicidad de Compuestos Fenólicos mediante QSAR", Pablo R. Duchowicz, Andrew G. Mercader, Eduardo A. Castro, Francisco M. Fernández
- III Congreso Iberoamericano de Ambiente y Calidad Vida, San Fernando del Valle, Catamarca, 25- 29 septiembre de 2006. Presentación del Poster: "ESTUDIO qsar DE LA HORMONA JUVENIL EN MOSQUITOS" P. R. Duchowicz; A. G. Mercader; M. Sanservino; E. A. Castro; F. M. Fernandez
- Eighth J. J. Giambiagi Winter School, Clusters, Molecules, Biomolecules and Materials, Buenos Aires, 24-28 Julio de 2006. Defensa del Poster: "QSPR Study on Fluorophilicity", Pablo R. Duchowicz, Andrew G. Mercader, Miguel A. Sanservino, Eduardo A. Castro, Francisco M. Fernández
- VIII Conference on Current Trends in Computational Chemistry (8th CCTCC), Vicksburg, Mississippi, USA, November 5-6, 1999. Comunicación: "Calculation of total electronic energies from correlation weighting of local graph invariants", A. Mercader, E. A. Castro y A. A. Toropov.

- Jornadas 2008 Becarios del INIFTA, "Nuevas aplicaciones en QSAR/QSPR", 10 de Octubre de 2008
- Jornadas 2007 Becarios del INIFTA, "Avances en el Desarrollo de Modelos QSAR/QSPR", 24 de Octubre de 2007
- Jornadas 2006 Becarios del INIFTA, "Experiencia laboral en Shell Argentina", 6 de Octubre de 2006
- 91° Reunión de la Asociación Física Argentina, Merlo, San Luis, 27 de septiembre de 2006. Presentación oral del trabajo: "Modelos de Regresión Lineal para la Estimación de Propiedades Fisicoquímicas"