Universidad Central "Marta Abreu" de Las Villas
Facultad de Matemática, Física y Computación

Caracterización de secuencias de ADN y proteínas a través del cálculo de los momentos espectrales

Trabajo de Diploma en Ciencia de la Computación

Autor: Antonio Bouza Pérez

Tutores: MSc. Guillermin Agüero Chapín

Dra. Leticia Arco García

Santa Clara, 2012

Por sobre todas las cosas a mi familia, a mamá, papá, mima y a mi hermano.

A la profesora Leticia por su guía y soportarme durante estos años.

Muy especiales a Guillermin por su apoyo y siempre estar dispuesto cuando lo necesitaba.

A la tía Olguita, a baby y a Monte.

A Evys, Reinaldo, Miguel Angel, Gisselle, Aliuska y demás compañeros del CBQ.

A Albanis y a Manuel.

RESUMEN

La caracterización de secuencias de ADN y proteínas mediante índices topológicos derivados de sus representaciones gráficas es un campo de investigación relativamente nuevo que ha tomado impulso en los últimos años. Sin embargo, existen pocos métodos libres de alineamiento en la Bioinformática que empleen enfoques gráficos para la caracterización matemática de secuencias de ADN y proteínas, así como aplicaciones que utilicen estas técnicas. Los software existentes solo hacen uso de una u otra representación gráfica de estas secuencias y a veces la información relevante aportada por los descriptores moleculares que calculan no es la que verdaderamente resulta de interés para los estudios que se acometen. De ahí que el objetivo de este trabajo es desarrollar un software para el análisis por métodos gráficos de secuencias de genes y proteínas que permita caracterizarlas numéricamente a través de índices topológicos. Los principales resultados obtenidos son: identificación de los principales métodos gráficos empleados en la representación de secuencias de ADN y proteínas; el software ESPECTRO que permite realizar un ciclo cerrado de análisis de las biomoléculas transitando desde cargar los datos en diferentes formatos, graficar las biomoléculas incluyendo varios tipos de representaciones, calcular los momentos espectrales a partir de la representación seleccionada y salvar los resultados para ser utilizados en otros sistemas que permitan realizar predicciones. Las representaciones y consecuentes caracterizaciones logradas con ESPECTRO aportan información útil que cuantifica la esencia de la composición y distribución de las secuencias proteicas que constituyen dominios de adenilación y en la predicción de funciones tipo bacteriocina.

Abstract

Tabla de contenidos

INTRODUCCIÓN

Al intentar caracterizar una secuencia de determinación reciente, queremos saber de qué proteína se trata, a qué familia puede pertenecer, cuál es su función biológica y cómo podemos explicar su función en términos estructurales. Todavía no existe la base de datos o el software que permita dar respuesta directa a todas estas cuestiones. Para dar solución a estas interrogantes es razonable conjuntar diversas técnicas en un protocolo de búsqueda. La mayoría de los métodos bioinformáticos que se emplean en el tratamiento de estos problemas basan su análisis en procedimientos de alineamiento de secuencias.

En términos simples el alineamiento de secuencias es el proceso en el cual diferentes secuencias son comparadas mediante la búsqueda de patrones comunes y el establecimiento de correspondencias residuo – residuo entre secuencias relacionadas. Los estudios comparativos que se realizan en los procedimientos de alineamientos de secuencias hacen uso de bases de datos públicas que se encuentran disponibles en Internet. Buscar en una base de datos biológica equivale a alinear la secuencia en estudio con las demás secuencias almacenadas, tratando de establecer un segmento entre ellas donde el número de coincidencias sea máximo.

Sin embargo, los métodos de alineamiento que se emplean en la caracterización de secuencias, tienden a actuar erráticamente cuando los miembros de la familia con la que se compara la secuencia analizada presentan divergencia en su estructura primaria. Además las metodologías de alineamiento múltiple de secuencias no son confiables para inferir relaciones evolutivas a bajos niveles de similitudes de secuencias.

La utilización de métodos gráficos es una herramienta útil para estudiar sistemas biológicos. Dichos métodos se han introducido además para los estudios de Relación Cuantitativa Estructura-Actividad (QSAR) y para tratar con complicados sistemas de redes(Gonzalez-Diaz, 2007).

Algunas representaciones gráficas de genes y proteínas basadas en sistemas cartesianos 2D han sido introducidas por Gates (Gates, 1985), Nandy (Nandy, 1994) y Leon – Mogenthaler (Leong and Mogenthaler, 1995). También recientemente Milan Randic (Randic and Balaban, 2005)ha hecho aportes en esta área, basado en estudios asociados a representaciones gráficas de genes y proteínas en mapas de colores.

Las representaciones gráficas de genes y proteínas permiten modelar la información contenida en largas cadenas de biopolímeros en estructuras pseudo – secundarias artificiales que son descritas a través de grafos. Luego, de estas estructuras pseudo – secundarias artificiales se tratan de derivar descriptores matemáticos que permitan inferir la actividad de la estructura real de la biomolécula. Los descriptores moleculares se obtienen considerando algunas características relevantes de las estructuras pseudo – secundarias artificiales como son la topología de los grafos y las propiedades fisicoquímicas de los monómeros que la integran. Los descriptores moleculares de este tipo son denominados índices topológicos.

La caracterización de secuencias de ADN y proteínas mediante índices topológicos derivados de sus representaciones gráficas es un campo de investigación relativamente nuevo que ha tomado impulso en los últimos años. En el grupo de Diseño de Fármacos y Simulaciones Moleculares del Centro de Bioactivos Químicos (CBQ) de la Universidad Central "Marta Abreu" de Las Villas se investiga sobre esta línea; ellos han extendido la aplicación de los índices topológicos definidos por Ernesto Estrada como los *momentos espectrales* asociados a la matriz de adyacencia de enlace de un grafo molecular al campo de las macromoléculas (Estrada, 1996, Estrada, 1997), específicamente en genes y proteínas.

La metodología empleada por los investigadores del CBQ para el análisis de secuencias de ADN y proteínas surge como complemento de las clásicas basadas en métodos estadísticos, que actúan sobre la base de procedimientos de alineamiento de secuencias. Para la experimentación y determinación de los dominios de aplicación donde la información aportada por la caracterización de secuencias de ADN y proteínas a través del cálculo de los *momentos espectrales* resulte de mayor utilidad, surge la herramienta (ESPECTRO *versión 1.0* ®).

Teniendo en cuenta las consideraciones anteriores se formula el siguiente **problema de investigación**. Existen pocos métodos libres de alineamiento en la bioinformática que empleen enfoques gráficos para la caracterización matemática de secuencias de ADN y proteínas, así como aplicaciones que utilicen estas técnicas. Los software existentes solo hacen uso de una u otra representación gráfica de estas secuencias y a veces la información relevante aportada por los descriptores moleculares que calculan no es la que verdaderamente resulta de interés para los estudios que se acometen.

De ahí que el **objetivo general** de este trabajo de diploma consiste en desarrollar un software para el análisis por métodos gráficos de secuencias de genes y proteínas que permita caracterizarlas numéricamente a través de índices topológicos.

Este se desglosa en los siguientes **objetivos específicos**:

- Identificar los principales métodos gráficos empleados en la representación de secuencias de ADN y proteínas.
- Diseñar e implementar un software que permita visualizar las principales representaciones gráficas realizadas para genes y proteínas.
- Permitir la lectura de los formatos en los cuales vienen representadas las secuencias de genes y proteínas (*.fasta, *.gb, *.gp, *.pdb).
- Caracterizar las biomoléculas mediante el cálculo de los *momentos espectrales.*
- Evaluar el desempeño de la herramienta y la calidad de la información aportada por los valores de los *momentos espectrales* que caracterizan las secuencias de genes y proteínas en modelos de clasificación.

Las **preguntas de Investigación** planteadas son:

- ¿Cuáles son las estructuras de datos más adecuadas para la implementación de los algoritmos asociados al cálculo de los *momentos espectrales* de secuencias de ADN y proteínas?
- ¿Qué plataforma es la más propicia para implementar las visualizaciones de las representaciones gráficas de genes y proteínas?
- ¿Qué estrategias implementar para el análisis eficiente de grandes cantidades de datos?

Hipótesis.

"El desarrollo de un software que permita representar, caracterizar y comparar biomoléculas siguiendo una metodología libre de alineamiento, le permite a un especialista en Bioinformática realizar un análisis integral de las biomoléculas utilizando representaciones gráficas y crea las bases para realizar futuras predicciones en esta área"

1 Biomoléculas y sus formas de representación basadas en la teoría de grafos

En este capítulo se presentarán los elementos fundamentales que caracterizan a las biomoléculas, haciendo énfasis en genes y proteínas. Se mostrarán algunas de las representaciones gráficas que se han creado para representarlas, entre ellas las cartesianas, las que representan al ADN como espectro, la representación de proteínas en grafos tipo estrella, las proteínas como matrices de adyacencia de aminoácidos y los mapas de mapas de cuatro colores. Para ello, primeramente se formalizarán elementos básicos de la teoría de grafos que se aplican en estas representaciones.

1.1 Biomoléculas

La composición química de los organismos vivos difiere considerablemente de la materia inanimada. Las moléculas que son específicas de los seres vivos son las biomoléculas; aunque es necesario aclarar que también forman parte de la materia viva algunas sustancias de naturaleza inorgánica.

Una característica esencial de las biomoléculas es su diversidad, a pesar de estar constituidas fundamentalmente por un grupo pequeño de átomos: carbono, nitrógeno, oxígeno, hidrógeno, y en menor cuantía fósforo y azufre, entre otros. La forma de asociarse dichos átomos, que explican la diversidad estructural de estas moléculas y de sus propiedades, depende de las características de sus elementos constituyentes, como la disposición en que éstos se unen para formar los enlaces y agrupaciones moleculares.

Las biomoléculas de mayor complejidad son las macromoléculas, biopolímeros formados por la unión de otras biomoléculas más sencillas, que constituyen sus monómeros o precursores. Las proteínas son polímeros de aminoácidos; los ácidos nucleicos, de nucleótidos y los polisacáridos, de monosacáridos (Cardellá and Hernández, 1999).

1.1.1 Proteínas

Las proteínas son las macromoléculas más abundantes de las células vivas, estando presente en todas las células y en todas las partes de las mismas. Éstas son las biomoléculas con mayor grado de variabilidad estructural, que desempeñan las funciones más diversas; muchas son enzimas, otras intervienen en el transporte de diferentes sustancias y constituyen los receptores de diversos ligandos; algunas forman los anticuerpos, varias son hormonas. Su papel central se manifiesta de

forma evidente en el hecho de que las proteínas son los productos finales más importantes de las rutas de información (Lehninger, 1987).

Las proteínas, sin excepción, están constituidas por la unión de moléculas sencillas llamadas aminoácidos, cuyo nombre alude a la presencia de un grupo amino, $-NH_2$, que le da carácter básico y un grupo carboxilo, -COOH, al que debe su carácter ácido (Lehninger, 1987).

Se han descubierto alrededor de veinte aminoácidos que forman parte de las moléculas proteicas (Kourí, 1978). Todos ellos poseen un átomo de carbono al que se unen, de modo constante, un grupo amino, un grupo carboxilo y un átomo de hidrógeno; por tanto, queda una valencia libre a la que se une un átomo o grupo de átomos diferentes para cada aminoácido, que se representa abreviadamente por R y significa radical, como se observa en la Figura 1.1.

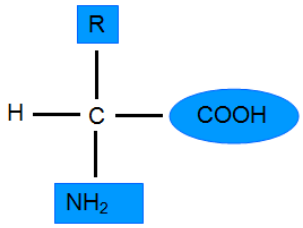


Figura 1.1. Fórmula general de un aminoácido.

El grupo R puede representar un átomo de hidrógeno, como sucede en la glicina, que es el aminoácido más sencillo; un grupo metilo, CH_3, como en la alanina; o combinaciones más complicadas en las que pueden estar presente átomos de azufre, por ejemplo, en la metionina, y con menos frecuencia, fósforo u otros elementos. Como se muestra en la Figura 1.2, también en estos radicales pueden estar presentes grupos amino, $-NH_2$, o carboxilo, -COOH.

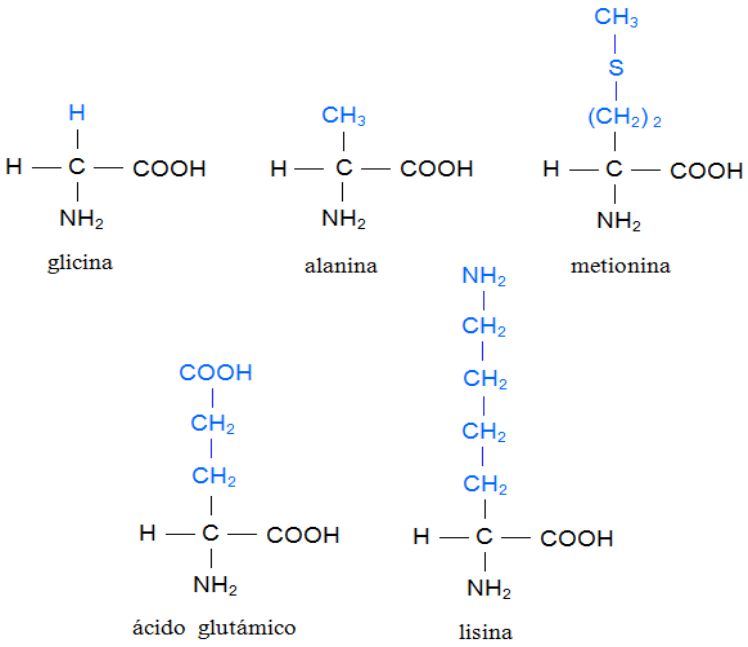


Figura 1.2. Fórmula de algunos aminoácidos. Lo indicado en azul representa los radicales característicos de diferentes tipos de aminoácidos.

Los aminoácidos se pueden clasificar sobre la base de distintos criterios, de acuerdo con el objetivo que se persiga. A continuación se presentan dos criterios de los más usados (Cardellá and Hernández, 1999):

- La cantidad de grupos carboxilos y aminos (u otra agrupación básica) presente en el aminoácido, de lo cual derivará el carácter ácido-básico de sus disoluciones. Sobre este criterio, los aminoácidos se clasificarán en neutros, ácidos y básicos. En la Tabla 1.1 se puede apreciar de manera fácil que los aminoácidos ácidos son dos (glutámico y aspártico), los básicos son tres (lisina, arginina e histidina), y el resto son aminoácidos neutros.

Neutros		Ácidos	Básicos
Glicina	Cisteína	Ácido glutámico	Lisina
Alanina	Metionina	Ácido aspártico	Arginina
Valina	Fenilalanina		Histidina
Leucina	Tirosina		
Isoleucina	Triptófano		
Asparagina	Prolina		
Glutamina	Treonina		
Serina			

Tabla 1.1. Clasificación de los aminoácidos según número de grupos carboxilos y aminos en la molécula.

- La presencia o no de grupos químicos polares en su cadena lateral R. Sobre este criterio, los aminoácidos se clasifican en apolares si no poseen ningún grupo polar en R, y polares -si tienen algún grupo polar en R. Los polares, a su vez, se subclasifican en polares iónicos -si a valores de pH fisiológico, adquieren carga eléctrica apreciable -y polares poco iónicos -si a valores de pH fisiológico, no adquieren carga eléctrica apreciable. En la Tabla 1.2 se muestra la ubicación de cada aminoácido de acuerdo con este fundamento de clasificación. Se puede observar que los polares iónicos son precisamente los dos aminoácidos ácidos y los tres básicos, según el criterio precedente de clasificación; son polares poco iónicos aquellos aminoácidos que presentan en R alguno de los grupos siguientes: hidroxilo (OH), sulfidrilo (SH), amida ($CONH_2$) o el anillo indol; el resto de los aminoácidos son apolares.

Polares		Apolares
Iónicos	**Poco iónicos**	Glicina
Ácido aspártico	Serina	Alanina
Ácido glutámico	Treonina	Valina
Lisina	Tirosina	Leucina
Arginina	Cisteína	Isoleucina
Histidina	Asparagina	Metionina
	Glutamina	Fenilalanina
	Triptófano	Prolina

Tabla 1.2. Clasificación de los aminoácidos según la polaridad de sus grupos R.

1.1.2 Ácidos nucleicos

Los ácidos nucleicos constituyen la segunda macromolécula de importancia biológica después de las proteínas, y sus funciones están muy relacionadas con estas últimas. Las funciones de los ácidos nucleicos están relacionadas con el funcionamiento del aparato genético celular, o sea, con el conjunto de moléculas y mecanismos que garantizan la trasmisión y expresión de los caracteres hereditarios de generación en generación y, por tanto, son de un gran valor en la perpetuación de las especies (Mathews et al., 1999).

Los dos tipos principales de ácidos nucleicos que se diferencian, tanto estructural como funcionalmente, son: los ácidos desoxirribonucleicos (ADN) y los ácidos ribonucleicos (ARN). De cada uno de ellos existen diferentes subtipos. Ambos surgen como consecuencia de la polimerización

de unidades estructurales más sencillas, denominadas nucleótidos. Estos están formados a su vez por tres componentes, como se muestra en la Figura 1.3: una base nitrogenada, un azúcar y un grupo fosfato.

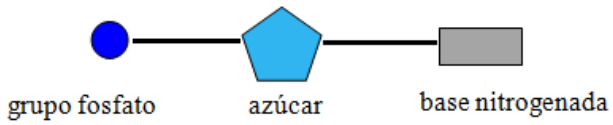


Figura 1.3. Esquema que representa los componentes de un nucleótido.

Las bases nitrogenadas son moléculas orgánicas constituidas por anillos que llevan en sus carbonos algunos grupos amino, que le dan carácter básico. Las bases que forman los ácidos nucleicos son de dos tipos: purínicas, cuando se originan por la sustitución de algunos átomos en un compuesto orgánico llamado purina, y pirimidínicas cuando se trata de otro, llamado pirimida (Lehninger, 1987).

Como se puede apreciar en la Figura 1.4, tanto el ADN como el ARN contienen dos bases purínicas principales, la **adenina** (A) y la **guanina** (G). El ADN y el ARN contienen también dos bases pirimidínicas principales; una de ellas es la **citosina** (C) en ambos tipos de ácido nucleico. La naturaleza de la segunda base pirimidínica es la única diferencia importante entre las bases del ADN y las de ARN: **timina** (T) en el ADN y **uracilo** (U) en el ARN.

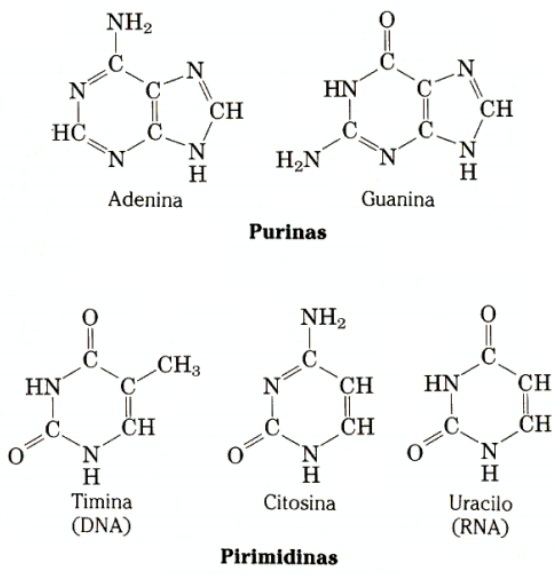


Figura 1.4. Esquema de las bases nitrogenadas de los ácidos nucleicos.

Estas bases se unen a un azúcar de tipo pentosa, ya sea ribosa en el caso del ARN o desoxirribosa para el caso del ADN. El compuesto formado por la unión de la base nitrogenada y la azúcar pentosa se denomina **nucleósido**.

En el ADN de cada célula se encuentran especificadas las secuencias de aminoácidos de todas sus proteínas y las secuencias de nucleótidos de todas sus moléculas de ARN. La información necesaria para construir las secuencias de nucleótidos del ARN o de proteínas se encuentra en las correspondientes secuencias de nucleótidos del ADN. Un segmento de ADN que contiene la información necesaria para la síntesis de un producto biológico funcional (proteína o ARN) recibe el nombre de **gen**. La única función del ADN es el almacenamiento de información biológica.

En la célula existen varias clases de ARN, cada uno de ellos con una función distinta. Los **ARN ribosómicos** (ARNr) son componentes estructurales de los ribosomas, complejos de gran tamaño que llevan a cabo la síntesis de proteínas. Los **ARN mensajeros** (ARNm) son ácidos nucleicos que transportan la información desde un gen o unos pocos genes hasta el ribosoma, donde se sintetizan las proteínas codificadas por dichos genes. Los **ARN de transferencia** (ARNt) son moléculas adaptadoras que traducen con fidelidad la información contenida en el ARNm en secuencias específicas de aminoácidos (Cardellá and Hernández, 1999).

Los métodos gráficos constituyen una herramienta útil para estudiar las biomoléculas (Nandy, 1994, Randic et al., 2010, Randic et al., 2006). Existen diversas formas de representación gráfica de los genes y proteínas. El uso de estas formas de representación posibilita que fragmentos con información relevante se puedan obtener rápidamente por la inspección visual de la trama de la secuencia. Estas modelaciones gráficas permiten establecer una relación entre la estructura de la molécula y su actividad, son libres de alineamientos, o sea, la evaluación de las similitudes en las cadenas de ARN y ADN no se basan en la comparación lineal de las secuencias de bases nitrogenadas que lo conforman, sino en variantes gráficas de la representación de la molécula. Por tal motivo, en el siguiente epígrafe se precisarán los conceptos principales de la teoría de grafos, que permitirán describir las representaciones gráficas más exitosas de genes y proteínas.

1.2 Conceptos principales de la teoría de grafos

La teoría de grafos ha sido utilizada para estudiar los problemas que surgen en una amplia variedad de áreas de aplicación incluyendo la Ciencia de la Computación, la Informática, la Ingeniería Eléctrica, la Química, la Sociología, las Ciencias Políticas y la Economía, sólo por nombrar unas pocas

(Balasubramanian, 1985, Semenovich, 2004, Venkata, 2007, McCall, 1985, Zhang et al., 2010). La Bioinformática no está exenta de utilizar la teoría de grafos como una herramienta poderosa para estudiar algunos de los problemas a resolver. Tal es el caso de la representación gráfica de genes y proteínas, donde los grafos juegan un rol fundamental. A continuación se establecen los conceptos, definiciones y algoritmos básicos de la teoría de grafos que son útiles al representar genes y proteínas.

1.2.1 Definiciones

Se dice **seudografo** $G = (V, A, f)$ a una terna, donde $V \neq \varnothing$ es el conjunto de nodos, puntos o vértices, A es el conjunto de aristas y $f: A \rightarrow V \times V \cup \{\{u,v\};\ u, v \in V\}$. $\forall a \in A$, si $f(a)=(u,v)$ o $f(a)=\{u,v\}$, se dice la arista a conecta los nodos u y v.

Si $G = (V, A, f)$ es un seudografo donde f es una función inyectiva, entonces la **notación** con frecuencia es de la forma $G = (V, A)$ o simplemente G. Cada arista se representa directamente como el par con el cual se corresponde. Cuando no sea necesario mencionar el conjunto de vértices y artistas, se puedo simplemente escribir G.

Sea el seudografo $G = (V, A, f)$, donde $a, b \in A$ y $u, v \in V$. Si $f(a) = (u,v)$ se dice que a es una **arista dirigida o arco**. Si $f(a) = \{u,v\}$ se dice que a es una **arista no dirigida**.

Los seudografos se pueden representar gráficamente mediante diagramas. La presentación gráfica de un seudografo consiste de un diagrama donde los vértices se representan por círculos, puntos, etc y las aristas por líneas que unen los vértices. Las líneas que representan arcos poseen flechas indicando la dirección.

Un elemento del conjunto $V \times V$ puede tener más de una pre-imagen en A, es decir, f puede no ser necesariamente inyectiva, por tanto, puede existir más de una arista que conecte dos vértices. En tal caso, si $f(a) = f(b) = (u,v)$ o $f(a) = f(b) = \{u,v\}$ se dice que a y b son **aristas paralelas** (o **aristas múltiples**). En las representaciones de genes y proteínas que se analizarán en este capítulo se utilizan seudografos que no tienen presencia de aristas múltiples, que son los llamados seudografos sencillos (o simples), donde la función f es inyectiva. Una arista a se dice **lazo** o **bucle** si $f(a) = (v,v)$ o $f(a) = \{v,v\}$. Los seudografos que se utilizan para representar genes y proteínas tampoco tienen presencia de lazos o bucles. De ahí que los seudografos a utilizar son los llamados **grafos**, que son aquellos seudografos sencillos sin lazos o bucles (Bondy and Murty, 1976).

En dependencia de las características de las aristas, existen diferentes tipos de seudografos: seudografo dirigido, seudografo no dirigido y seudografo mixto. En este trabajo se aplicarán seudografos no dirigidos. Un seudografo se dice **no dirigido o no orientado** si todas sus aristas son no dirigidas.

Dos **aristas** se dicen **adyacentes** si tienen un vértice común. Si *a* es una arista asociada a los vértices *u* y *v*, los vértices *u* y *v* se llaman **vértices extremos** de la arista *a*; *u* y *a* son **incidentes** (también lo son *v* y *a*); los vértices *u* y *v* son **adyacentes.**

Un seudografo se dice **ponderado** si cada una de sus aristas tiene asignada una etiqueta. En dependencia de la aplicación, esta etiqueta puede representar un determinado valor, peso, costo, longitud, etc. También existen seudografos donde es necesario ponderar los vértices.

Un concepto muy utilizado cuando se aplica la teoría de grafos es el de **grado de un vértice**. El **grado** de un vértice en un seudografo no dirigido es la cantidad de aristas que son incidentes a él. Si *v* es un nodo de un seudografo no dirigido, el grado de *v* se denota por $\delta(v)$.

Frecuentemente es necesario calcular la similitud entre cadenas de ARN y ADN, y cuando estas biomoléculas se representan utilizando grafos, las similitudes no se basan en la comparación lineal de las secuencias de bases nitrogenadas que las conforman, entonces un concepto importante para comparar los grafos utilizados en la modelación es el isomorfismo de grafos.

Sean $G = (V, A)$ y $G' = (V', A')$ seudografos no dirigidos. Se dice que *G* y *G*' son **isomorfos** sí y solo sí existen dos biyecciones $f: V \rightarrow V'$ y $F: A \rightarrow A'$ tales que $F(\{u,v\}) = \{f(u), f(v)\}$. Cuando dos seudografos son isomorfos, existe una completa correspondencia entre sus vértices y aristas de modo que ambos poseen las mismas propiedades.

El isomorfismo de seudografos es una relación de equivalencia. Como consecuencia, la relación de isomorfismo de seudografos determina una partición del conjunto de todos los seudografos en clases de equivalencia, donde a cada clase pertenecen todos los seudografos isomorfos entre sí. El concepto de isomorfismo sirve para establecer la igualdad o no entre seudografos arbitrarios. De esta manera, en cada clase de equivalencia por isomorfismo se puede señalar a un solo seudografo representante de su clase, ya que todos los demás poseen las mismas características (Harary, 1969).

En un seudografo no dirigido se llama **camino** (o **cadena**, o **ruta**) a una sucesión de aristas, siendo estas adyacentes consecutivas si son más de una. Si $a_1, a_2, a_3, ..., a_n$ es una sucesión, se acostumbra a escribir $(a_1, a_2, a_3, ..., a_n)$. Los vértices de un camino son aquellos que son incidentes a las aristas que

forman el camino. En un grafo no dirigido un camino se puede considerar como una sucesión de vértices que son consecutivamente adyacentes, dichos vértices son incidentes a las aristas que conforman el camino. Si v_1, v_2, v_3,..., v_n es una sucesión de vértices, se acostumbra a escribir (v_1, v_2, v_3,..., v_n). Si el camino tiene una única arista se escribe (v_1, v_2).

Se llama **camino simple** a aquel camino en el que todas sus aristas son diferentes, y **elemental** a aquel camino en el que todos sus vértices son diferentes. Todo camino contiene un camino simple. Un camino en el que el vértice inicial y final coinciden se llama **camino cerrado** o **ciclo**. En un camino cerrado o ciclo puede haber coincidencias de otros vértices además de los extremos, inclusive coincidencia de aristas. El ciclo se llama **simple** si ninguna arista del ciclo aparece más de una vez en el camino y **elemental** si solo coinciden el vértice inicial y final. Un seudografo no dirigido que no contiene ningún ciclo se denomina **acíclico.**

Un concepto que se utiliza mucho al representar biomoléculas con grafos es de **longitud** de un camino, que se refiere a la cantidad de aristas que conforman el camino. Sea un seudografo no dirigido y *u*, *v* vértices alcanzables uno a partir del otro (es decir, existe algún camino de uno al otro), se llama **distancia** de *u* a *v* a la longitud de un camino de longitud mínima de *u* a *v*, y se denota por ***d*(*u*,*v*).**

La relación *R* definida sobre el conjunto de vértices *V* de un seudografo no dirigido, donde $u, v \in V$; $uRv \Leftrightarrow$ existe un camino de *u* a *v*, se denomina **relación de camino**. La relación *R* es una relación de equivalencia sobre los vértices de un seudografo no dirigido. Así, el conjunto de vértices *V* de *G* queda particionado en clases de equivalencia. Dos vértices están en la misma clase si pueden unirse por un camino y en clases distintas en caso contrario.

Sea $G = (V, A)$ un seudografo no dirigido y $V_1, V_2, \ldots, V_r$ la partición de *V* según *R*. Sea A_i $(1 \leq i \leq r)$ el subconjunto de *A* formado por las aristas cuyos extremos están ambos en V_i. Los seudografos $G_i = (V_i, A_i)$ se conocen como las **componentes** de *G*. Un seudografo no dirigido *G* es **conexo** si tiene un único componente. Los grafos que se utilizarán para representar las biomoléculas son grafos conexos, por tanto, para cualquier par de vértices existe un camino que los une. En un seudografo no dirigido conexo *G*, se llama **diámetro** de *G* a la mayor de las distancias entre vértices que existe, y se denota por $D(G)$ (Harary, 1969).

1.2.2 Representación de seudografos

Sólo es posible realizar una representación de los grafos en forma de diagrama cuando el número de vértices y de aristas es razonablemente pequeño. Existen métodos alternativos para la representación de seudografos, por ejemplo, utilizando listas o matrices (Jungnickel, 2008).

1.2.2.1 Matriz de adyacencia

Sea $G = (V, A)$ un grafo dirigido en el cual $V = \{v_1, v_2, \ldots, v_n\}$ y se supone que los vértices están ordenados desde v_1 hasta v_n. La matriz $Y(G)$, nxn, cuyos elementos y_{ij} están dados por

$$y_{ij} = \begin{cases} 1 & \text{si} \quad (v_i, v_j) \in A \\ 0 & \text{si} \quad (v_i, v_j) \notin A \end{cases} \tag{1.1}$$

se denomina **matriz de adyacencias** del grafo G.

Para un grafo dirigido $G = (V, A)$, la matriz de adyacencias depende del orden de los elementos de V. Para distintos órdenes de los elementos de V, se obtienen distintas matrices de adyacencias de un mismo grafo G. Sin embargo, cualquiera de las matrices de adyacencia de G se puede obtener a partir de otra matriz de adyacencias del mismo grafo, sin más que intercambiar algunas de las filas y las columnas correspondientes de la matriz. Se tomará cualquier matriz de adyacencia del grafo como la matriz de adyacencias de ese grafo.

Teoremas importantes de esta representación son:

- El número de caminos de longitud λ desde v_i a v_j se obtiene por la posición i, j de la matriz potencia $[Y(G)]^{\lambda}$.
- Una condición necesaria y suficiente para que en G no existan ciclos es que para cierto valor λ suficientemente grande, $[Y(G)]^{\lambda} = 0$.

Los seudografos no dirigidos también se pueden representar con matrices de adyacencia, nótese que estás matrices serían siempre simétrica, solo sería necesario trabajar con una de las triangulares de dicha matriz. Las matrices de adyacencias no son muy eficientes para representar seudografos no dirigidos porque la información aparece dos veces (excepto aquella que aparece en la diagonal principal). Estas matrices también se pueden utilizar para representar seudografos ponderados, sustituyendo la presencia o no de aristas entre pares de vértices, por el valor de la ponderación de la arista.

1.2.2.2 Matriz de caminos

Sea $G = (V, A)$ un grafo dirigido en el cual $|V| = n$ y se supone que los vértices de V están ordenados. La matriz $P(G)$, nxn, cuyos elementos están dados por:

$$p_{ij} = \begin{cases} 1 & \text{si existe un camino desde } v_i \text{ hasta } v_j \\ 0 & \text{en caso contrario} \end{cases} \quad (1.2)$$

es denotada **matriz de caminos** (o **matriz de alcanzabilidad**) del grafo G.

Un elemento de la diagonal principal p_{ii} es igual a 1 si y sólo si existe un camino que vaya desde v_i hasta sí mismo, es decir, un ciclo.

Obsérvese que la matriz de caminos sólo muestra la presencia o ausencia de al menos un camino entre un par de vértices, y también la presencia o ausencia de ciclos en cualquier vértice. Sin embargo, no muestra todos los caminos que pudieran existir. En este sentido, una matriz de caminos no proporciona una información completa acerca de un grafo, tal como lo hace la matriz de adyacencias.

Un método eficiente para calcular la matriz de caminos lo constituye el Algoritmo de Warshall. (Harary, 1969)

1.2.2.3 Matriz de incidencias

Sea $G = (V, A)$ un seudografo dirigido, se llama **matriz de incidencias** a la matriz $E(G) = (e_{ij})$.

$$\text{Donde } e_{ij} = \begin{cases} -1 & \text{si el vértice } v_i \text{ es comienzo del arco } a_j \\ 1 & \text{si el vértice } v_i \text{ es final del arco } a_j \\ 0 & \text{si el vértice } v_i \text{ no es incidente al arco } a_j \end{cases} \quad (1.3)$$

Esta matriz, a diferencia de las matrices de adyacencias permite definir completamente un seudografo dirigido o no dirigido, ya que al representarse cada arista, quedan bien representados las aristas múltiples y los bucles. Al representar los seudografos no dirigidos con esta matriz, no es necesario distinguir entre vértices de entrada y de salida porque las aristas son no dirigidas, es suficiente especificar qué vértices inciden en cada arista. No es eficiente utilizar esta representación en grafos no dirigidos, donde no existen aristas múltiples ni bucles y por tanto la función f es inyectiva.

1.2.2.4 Listas de adyacencias

Sea $G = (V, A)$ un grafo dirigido, la lista que se forma para cada $v \in V$ y que contiene para v todos aquellos vértices u tales que $(v, u) \in A$, se denomina **lista de adyacencias**.

Los grafos no dirigidos también se pueden representar utilizando listas de adyacencias; sin embargo, cada arista se representará dos veces, una en cada dirección.

La representación del grafo en forma de lista de adyacencia suele ser preferible cuando el grafo es disperso, esto es, cuando para cada nodo hay tan sólo unas pocas aristas que inciden a él. Por otra parte, puede ser más adecuada una representación en forma de matriz de adyacencia si el grafo es denso.

Las listas de adyacencias no permiten definir completamente los multigrafos, a no ser que se hagan modificaciones para representar las aristas múltiples. Esta desventaja no limita su uso en la aplicación que se aborda en esta investigación.

Esta representación también es fácilmente adaptada a grafos ponderados añadiendo un único campo de datos a cada elemento de la lista para almacenar el peso asociado con esa arista. El número total de punteros a listas utilizados en esta representación es $|V|$ y la suma de las longitudes de todas las listas de adyacencias es $|E|$ en un grafo dirigido y $2|E|$ en un grafo no dirigido – para cada arista (u, v) de un grafo no dirigido la etiqueta de u aparece en la lista de v y la etiqueta de v aparece en la lista de u. Así, la memoria requerida para la representación con listas de adyacencia de un grafo $G = (V, E)$ es $\theta(\max(V, E)) = \theta(V + E)$.

1.2.3 Técnicas de búsqueda

En la gran mayoría de los problemas de grafos, a veces se hace necesario examinar todas las aristas y nodos que conforman el mismo. Una forma sistemática de examinar un grafo emplea un algoritmo de búsqueda que mantiene un conjunto $S \subseteq V$ de nodos de la siguiente manera: Inicialmente todos los nodos de V están sin marcar, excepto un nodo origen s y $S = \{s\}$. En cada iteración del algoritmo, un nodo u se quita de S y se procesa como sigue: Para cada arista (u, v) incidente desde u, si el nodo v está sin marcar, entonces se marca y se añade a S. Este proceso continúa hasta que S se hace vacío, en cuyo momento cualquier nodo que permanezca sin marcar no puede ser accedido desde un camino que comience en s. Posteriormente, puede seleccionarse un nuevo nodo origen entre los

nodos restantes sin marcar y el proceso de búsqueda puede continuar hasta que no queden nodos sin marcar.

La única parte del algoritmo de búsqueda arriba mencionado no completamente especificada es el orden en el cual los nodos se eliminan de *S*. Si se trata a *S* como una cola, quitando el nodo que ha permanecido más tiempo en *S*, entonces los nodos de *V* serán examinados desde *s* en anchura. Esto es, todos los nodos adyacentes a *s* serán examinados primero, después serán examinados los nodos adyacentes a éstos, y así sucesivamente. Por el contrario si *S* es tratado como una pila, siempre quitando el nodo añadido más recientemente, entonces los nodos de *V* serán examinados desde *s* en profundidad. Estos dos enfoques básicos conducen a los algoritmos 1.1 y 1.2 (Heileman, 1998).

```
Anchura (GRAFO G = (V, E), nodo s)          ► s es el origen
1   para cada v∈V hacer
2      d[v] ← false                          ► todos los nodos están no marcados
3   d[s] ← true                              ► marcar el origen
4   Añadir (Q, s)
5   mientras Vacía(Q) = falso  hacer
6      v ←  Avanzar (Q)
7      para cada u ∈ adyacentes[v]  hacer
8         si  d[u] = falso  entonces         ► ¿está marcado el nodo u?
9            d[u] ← true                     ► marcar el nodo u
10           Añadir(Q, u)
```

Algoritmo 1.1. Búsqueda en anchura en un grafo

```
Profundidad  (GRAFO G = (V, E))
1    para cada v∈V  hacer
2       d[v] ← false                         ► todos los nodos están sin descubrir
3    para cada s∈V  hacer
4       si  d[s] = false  entonces           ► ¿está marcado el nodo s?
5         Visita (G, s)


Visita (GRAFO G = (V, E), nodo s)            ► s es el origen
1    d[s]  = true                            ► el nodo s es descubierto y marcado
2    para cada u ∈ adyacentes[s]  hacer
3       si  d[u] = false  entonces           ► ¿está marcado el nodo u?
4         Visita (G, u)
```

Algoritmo 1.2. Búsqueda en profundidad en un grafo

En ambos algoritmos se utiliza un arreglo booleano `d[1..|V|]` para mantener los nodos marcados y no marcados de *V*. Con este esquema, un nodo *i* se considera no marcado si `d[i]` almacena el valor `false`, y marcado si `d[i]` es igual a `true`.

En el Algoritmo 1.1 en las líneas 1-3 se lleva a cabo la tarea de marcar el nodo origen y de designar inicialmente todos los nodos como no marcados. En el resto del algoritmo, se quita de la cola un nodo cada vez, comenzando con el origen *s*. Para cada nodo *v* quitado, todo nodo *u* adyacente y no marcado es primero marcado en la línea 9 y después insertado en la cola en la línea 10.

En el **Algoritmo 1.2** mostrado, en vez de emplear explícitamente una pila, se utiliza recursión para asegurar que los nodos marcados son considerados en orden LIFO. Debido a que es una búsqueda en profundidad, una vez que un nodo origen *s* es seleccionado, el algoritmo buscará en el grafo con mayor profundidad (desde *s*) hasta que ya no pueda encontrar un nodo no marcado. En este punto, si hay nodos restantes no marcados, se selecciona uno de ellos como el nuevo origen para seguir buscando. En el **Algoritmo 1.2** el método *Profundidad()* es el responsable inicialmente de hacer que todos los nodos estén no marcados y de escoger los nodos origen. En *Profundidad()* se invoca al método *Visita()* para realizar la búsqueda en profundidad desde cada origen. Las llamadas recursivas se hacen desde *Visita()* en la línea 4, devolviendo el control a Profundidad() sólo después de que todos los nodos no marcados accesibles desde un origen *s* hayan sido marcados.

Ambos algoritmos son aplicables en grafos no dirigidos, así como en grafos ponderados. En una representación del grafo de entrada como lista de adyacencia el tiempo de ejecución de estos dos algoritmos es $\theta(V + E)$.

1.3 Formas de representación gráfica de ADN, ARN y proteínas

La utilización de métodos gráficos es una herramienta útil para estudiar sistemas biológicos. Para el caso de los genes y proteínas existen distintas formas de representación gráfica de las mismas. El uso de éstas en genes y proteínas posibilita que fragmentos con información relevante puedan ser obtenidos rápidamente por la inspección visual de la trama de la secuencia. Existen diferentes técnicas de ploteo de secuencias de ADN y proteínas que abarcan desde los espacios 2-D, 3-D, hasta espacios complejos de cinco y seis dimensiones. A continuación se describirán las principales representaciones gráficas de genes y proteínas en espacios 2-D (Nandy et al., 2006, Randic et al., 2010).

1.3.1 Representaciones cartesianas

Las representaciones basadas en un Sistema Cartesiano de dos dimensiones sigue siendo la forma principal de métodos gráficos por su simplicidad y percepción intuitiva. Para el caso del ADN, las secuencias de nucleótidos son ploteadas en un camino a través de una cuadrícula, usando cuatro direcciones cardinales representadas cada una por una base. La idea es leer una secuencia de ADN base por base e ir ploteando puntos en un grafo. La ruta que tome el camino va a estar determinada en dependencia de cómo se asocien las cuatro direcciones cardinales a cada una de las bases. En (Nandy, 1994) un punto será ploteado moviéndose un paso en la dirección del eje x-negativo si la base es adenina (A), en dirección opuesta si la base fue guanina (G); se avanzará un paso en la dirección del eje y-positivo si la base fuera citosina (C) y en dirección opuesta corresponderá si la base fuera timina (T). (Gates, 1985) y (Leong and Mogenthaler, 1995) han propuesto otras variantes de asociar cada base a cada una de las cuatro direcciones cardinales. En la Figura 1.5 se resume las ideas expuestas por Gates, Nandy y Leon-Mogenthaler en sus trabajos.

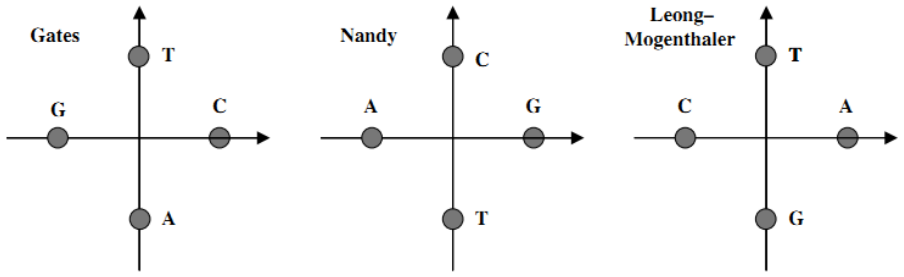


Figura 1.5. Representaciones gráficas para secuencias de ADN en sistemas cartesianos 2D

Así, una traza como ATGGTGCACC desplegará en los tres sistemas una traza como la mostrada en la Figura 1.6.

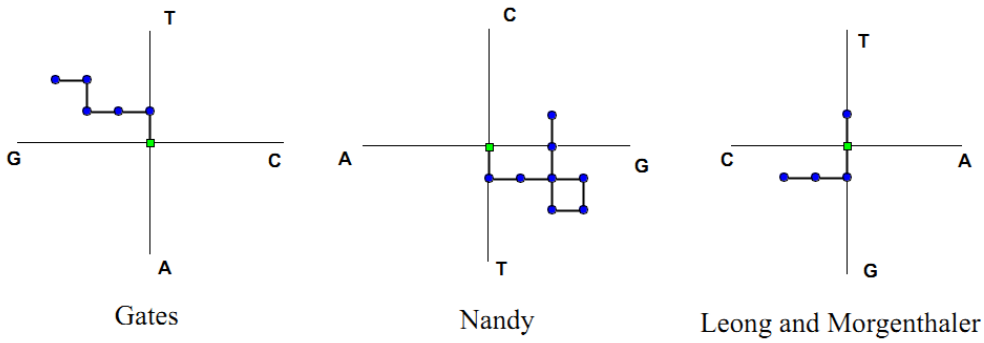


Figura 1.6. Representación cartesiana de la secuencia ATGGTGCACC

Es interesante notar que estos tres sistemas de coordenadas son exhaustivos en cuanto a las posibles representaciones de cuatro bases en un espacio 2-D, por lo que unidos forman un completo conjunto de descripciones para una secuencia dada.

La limitación inherente a estos trazados rectangulares es que en su trayectoria puede existir solapamiento de caminos. Esta degeneración produce pérdida de información, lo que trae consigo que las representaciones gráficas no sean necesariamente únicas y que de la trama del ADN graficado no sea posible la reconstrucción de la secuencia inicial. Sin embargo, del punto de vista práctico, es improbable que, para las secuencias de ADN de interés, uno arribe a idénticas representaciones gráficas partiendo de dos secuencias diferentes de ADN, particularmente para secuencias de moderado tamaño. Es más, aún cuando tal caso ocurriera, esto no significa que la representación gráfica perdiera valor, pero puede quizás hacerlas más interesante todavía. Esto es debido a que las representaciones gráficas fueron introducidas para facilitar la búsqueda de secuencias similares de ADN; y secuencias de ADN que tengan la misma representación gráfica pueden, indudablemente, tener similitudes inherentes.

La representación cartesiana hecha para secuencias de ADN, se puede extender también a proteínas. Para ello el conjunto de 20 aminoácidos que conforman las proteínas se divide en cuatro subgrupos. Cada subgrupo lo integraran aminoácidos que compartan propiedades afines. La división más común es en aminoácidos polares, apolares, básicos y ácidos. La composición de cada uno de estos subgrupos se muestra en la Tabla 1.3.

Apolares	Polares	Ácidos	Básicos
Glicina	Serina	Ácido glutámico	Lisina
Alanina	Treonina	Ácido aspártico	Arginina
Valina	Tirosina		Histidina
Leucina	Cisteína		
Isoleucina	Glutamina		
Metionina	Asparagina		
Fenilalanina	Triptófano		
Prolina			

Tabla 1.3. Aminoácidos polares, apolares, básicos y ácidos.

Luego cada subgrupo de aminoácido es asociado a una dirección del plano cartesiano. En la herramienta ESPECTRO, que será descrita en el Capítulo 3, la distribución que se emplea es la mostrada en la Figura 1.7.

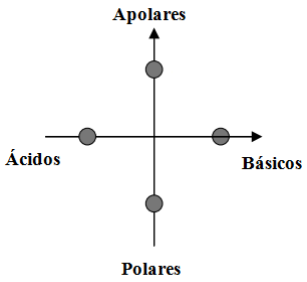


Figura 1.7. Direcciones del plano cartesiano asociadas a cada subgrupo de aminoácidos.

1.3.2 Representación del ADN como espectro

En esta representación, se pintan cuatro líneas en una superficie y se etiquetan con A, G, T y C. La secuencia de ADN estudiada se coloca debajo de las cuatro líneas, estando separadas las bases que la conforman a una distancia uniforme. Para obtener la representación gráfica del ADN, se trazan líneas través de la secuencia de ADN y se asigna a cada base un punto en la correspondiente línea horizontal. Los puntos así distribuidos en las cuatro líneas son conectados sucesivamente, formando un espectro que caracteriza la secuencia de ADN. Una representación de este espectro se muestra en la Figura 1.8.

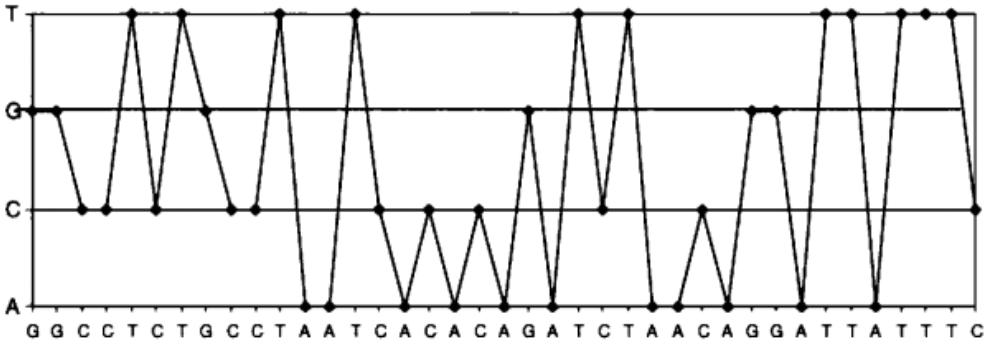


Figura 1.8. Representación de una secuencia de ADN como espectro.

Este método es útil pues no tiene degeneración involucrada. Las cuatro líneas horizontales pueden ser etiquetadas en cualquier orden, por lo que habrá 4! = 24 posibles grafos asociados a cada secuencia de ADN. Entre aquellos que usan este método se encuentra Randic, Vracko, Lers and Plavsic (Randic et al., 2003) que utilizaron la técnica en problemas de alineamiento de secuencias de ADN y en general, en estudios comparativos de ADN. Normalmente se remplazan las letras A, C, G, T que etiquetan las cuatro líneas por los números 1, 2, 3, 4; esto transforma la secuencia *alfabética* en una secuencia *numérica*. La ventaja de esta "trivial" substitución es que sobre la secuencia numérica se permite realizar ahora operaciones aritméticas que pueden presentarse gráficamente y pueden visualizarse.

1.3.3 Representaciones de proteínas en grafos tipo estrella

Los grafos estrella están conformados por un vértice central y numerosas ramas en formas de rayos, en los cuales solo aparecen vértices de grado dos o uno. En la Figura 1.9 aparecen en tres formas diferentes grafos estrella que tienen 11 ramas y 21 vértices además del vértice central.

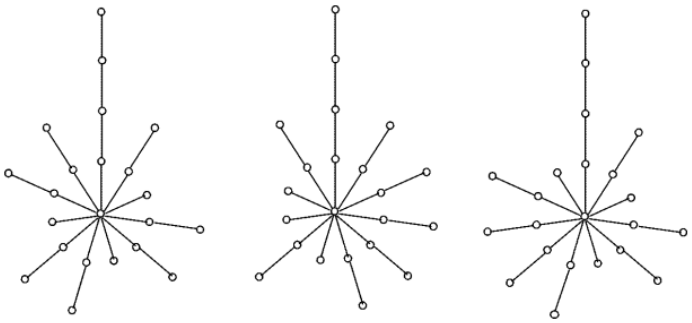


Figura 1.9. Grafos tipo estrella

A pesar de que lucen diferentes los diagramas, las tres formas representan la misma secuencia, por tanto son grafos isomorfos. Ni las longitudes de las aristas, ni su posicionamiento relativo son importantes, con tal que se mantenga la adecuada conectividad del grafo.

¿Cómo se asocia un grafo tipo estrella a una proteína? Se hace corresponder cada rayo con un aminoácido del conjunto de los 20 que forman las proteínas, siendo el número de vértices que integran el rayo la cantidad de veces que aparece ese aminoácido en la secuencia que describe la proteína.

El grafo de la Figura 1.9 permite (3!) * (7!) = 30240 diferentes asignaciones de aminoácidos a sus vértices conociendo que tres aminoácidos ocurren una sola vez, que siete aminoácidos ocurren dos veces y que solo un aminoácido se repite cuatro veces. De este mismo modo el grafo anterior se puede hacer corresponder a más de 30000 proteínas. Esto representa una pérdida colosal de información que típicamente acompaña a varias representaciones gráficas de biosecuencias. Sin embargo, esta pérdida de información puede ser recuperada cuando se asignan etiquetas a los vértices del grafo como se mostrará a continuación.

El grafo de la Figura 1.10 es la representación gráfica de la cadena A de la insulina humana que involucra a los siguientes 21 aminoácidos:

Gly-Ile-Val-Glu-Gln-Cys-Cys-Thr-Ser-Ile-Cys-Ser-Leu-Tyr-Gln-Leu-Glu-Asn-Tyr-Cys-Asn

Se puede apreciar que Cys aparece cuatro veces, Asn, Gln, Glu, Ile, Leu, Ser y Tyr aparecen dos veces, mientras que Gly, Thr y Val aparecen solo una vez. En la Figura 1.10 (izquierda) se ha realizado el diagrama del grafo etiquetando cada una de las 11 ramas con los aminoácidos que describen la secuencia de la cadena A de insulina humana. Se empieza con el aminoácido Gly y se termina con Asn. Es importante hacer notar que el vértice central de la estrella, el cual representa su origen, no pertenece a ningún aminoácido particular. En la Figura 1.10 (derecha) se realiza otro diagrama del mismo grafo, pero esta vez las etiquetas del 1 – 21 que se añaden a sus vértices indican la posición individual de cada aminoácido en la secuencia.

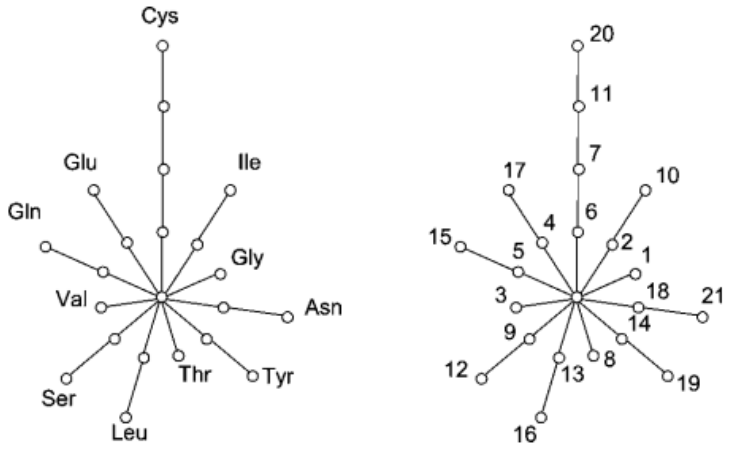


Figura 1.10. Representación de la cadena A de la insulina humana en un grafo tipo estrella.

1.3.4 Proteínas como matrices de adyacencia de aminoácidos

Se puede asociar a cada proteína una matriz de 20*20 (AA), donde cada fila y columna representa uno de los 20 aminoácidos que conforman las proteínas. El elemento (*i, j*) de la matriz indicará el número de veces que el aminoácido (*i*) es seguido a continuación por el aminoácido (*j*) en la secuencia de proteína cuando esta es leída de izquierda a derecha. Esta matriz no es simétrica, y cuando un aminoácido se repite seguidamente en la secuencia, se incrementa su valor en la diagonal principal. La suma de los elementos de una fila de esta matriz proporciona la abundancia del correspondiente aminoácido en la secuencia, exceptuando el último aminoácido de la secuencia, el cual no es incluido en su fila. Igualmente, la suma de una columna proporciona la abundancia de dicho aminoácido en la secuencia, a excepción del primero que no se incluye en su correspondiente suma. Mientras que los valores en las filas indican las adyacencias cuando la proteína es leída de izquierda a derecha, las entradas en la columna indican las adyacencias cuando la secuencia de proteína es leída de derecha a izquierda.

Resulta de interés la construcción de matrices simétricas de adyacencias de aminoácidos, estas se pueden obtener adicionando la matriz de adyacencia AA con su traspuesta AA^T. La secuencia:

WTFESRNDPAKDPVILWLNGGPGCSSLTGL

perteneciente a la levadura *Saccharomyces cerevisiae* se representa en la Figura 1.11 mediante el grafo asociado a la matriz AA de dicha secuencia.

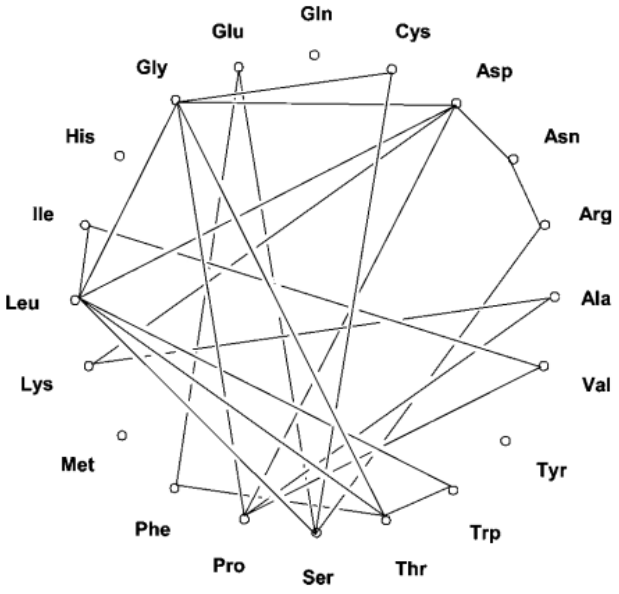


Figura 1.11. Representación de la proteína *Saccharomyces cerevisiae* que describe las conexiones locales de sus aminoácidos

1.3.5 Mapas de cuatro colores

La representación de biomoléculas en mapas de colores es aplicable tanto a genes como a proteínas. Primeramente, se muestra un ejemplo de su uso en secuencias de ADN, tomando de muestra el exón 1 del gen de B-hemoglobina humana. El primer paso es plegar la secuencia en forma de espiral dentro de una cuadrícula como se muestra en la Figura 1.12 (a). Se ha hecho énfasis en una arista de la espiral para una mejor visibilidad.

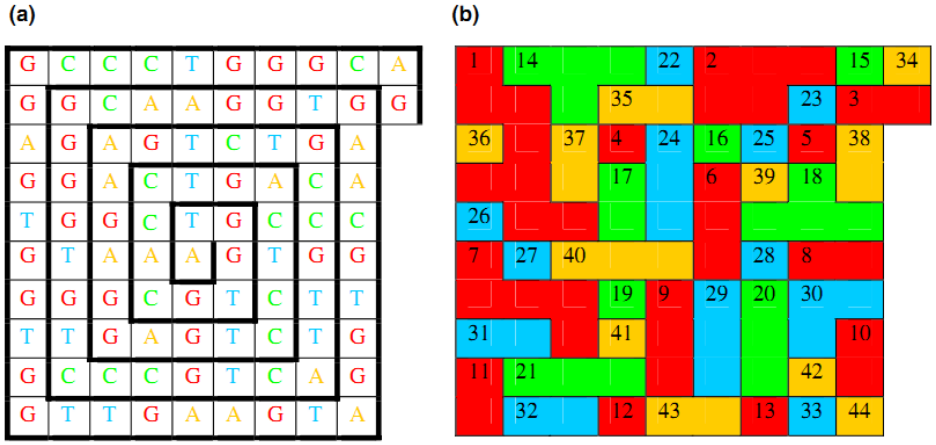


Figura 1.12. (a) Secuencia del exón 1 del gen de B-hemoglobina humana codificada dentro de una cuadrícula en forma de espiral. (b) Representación en mapas de colores de la misma secuencia.

Como son cuatro los nucleótidos que conforman el ADN, se emplean cuatro colores para diferenciar a cada uno. La espiral empieza en el cuadro centro de la cuadrícula que contiene el nucleótido A (adenina), y termina en la última celda de la periferia que contiene a una G (guanina). El resultado es una cuadrícula etiquetada con cuatro letras A, C, G y T cada una de un color distinto. El siguiente paso es agrupar las celdas adyacentes que presenten igual color. Dos celdas son adyacentes si tienen una arista común. En la Figura 1.12 (b) se puede apreciar el mapa de cuatro colores conformado de la secuencia ejemplo.

Esta representación permite hacer una inspección visual de similitudes/disimilitudes entre secuencias de ADN. Es importante señalar que pequeños cambios en la secuencia de nucleótidos pueden alterar considerablemente la configuración final del mapa de cuatro colores.

En secuencias de proteínas igualmente se pliega la cadena en forma de espiral sobre una cuadrícula. Los principales cambios respecto al modo de operar en secuencias de ADN, son referidos al aspecto de agrupar las celdas, pues son diferentes los enfoques que se emplean. Como el alfabeto de las proteínas lo integran 20 tipos de aminoácido, la probabilidad de que dos aminoácidos del mismo tipo sean adyacentes se reduce, comparados con la probabilidad de agrupar dos nucleótidos de un conjunto de cuatro para el caso de la cuadrícula que se obtiene de la representación de un gen. Debido a esto es frecuente dividir el conjunto de 20 aminoácidos en subgrupos por propiedades afines. La división más común es en aminoácidos polares, apolares, básicos y ácidos, como se muestra en la Tabla 1.3. Luego, el mapa será de tantos colores como subgrupos se hayan conformado. Dos celdas adyacentes se agrupan si los aminoácidos que la integran pertenecen a un mismo subgrupo. Esta es una posible opción que se sigue, sin embargo, se puede crear un mapa de 20 colores, donde solo dos celdas adyacentes se agrupan si contienen el mismo aminoácido.

1.4 Consideraciones finales del capítulo

En este capítulo se han descrito los elementos fundamentales que caracterizan a las biomoléculas, particularizando en dos tipos principales de ácidos nucleicos que se diferencian, tanto estructural como funcionalmente, son: los ácidos desoxirribonucleicos (ADN) y los ácidos ribonucleicos (ARN); y haciendo énfasis principal en las proteínas.

Se mostraron algunas de las representaciones gráficas que se han creado para representar genes y proteínas, entre ellas las cartesianas, las que representan al ADN como espectro, la representación de proteínas en grafos tipo estrella, las proteínas como matrices de adyacencia de aminoácidos y los mapas de cuatro colores. Cada una de estas representaciones permite extraer información de las secuencias desde distintos enfoques, algunas solo aplicables a proteínas y otras solo a ADN o ARN, y otras generalizables. El uso de estas formas de representación posibilita que fragmentos con información relevante se puedan obtener rápidamente por la inspección visual de la trama de la secuencia, permiten establecer una relación entre la estructura de la molécula y su actividad, son libres de alineamientos, o sea, la evaluación de las similitudes en las cadenas de ARN y ADN no se basan en la comparación lineal de las secuencias de bases nitrogenadas que lo conforman, sino en variantes gráficas de la representación de la molécula.

Elementos básicos de la teoría de grafos fueron mostrados para esclarecer los tipos de seudografos que se utilizan así como los principales algoritmos que se aplican después de realizar las

representaciones. La mayoría de las modelaciones utilizan grafos no dirigidos, en algunos casos ponderados, representados en matrices y listas de adyacencias, sobre las cuales se aplican recorridos a lo ancho y en profundidad.

Las representaciones gráficas ofrecen posibilidades visuales para el análisis de los biomoléculas, sin embargo, su mayor potencialidad consiste en sentar las bases para caracterizarlas numéricamente y así permitir el cálculo de sus similitudes a partir de descriptores que se identifiquen.

2 Caracterización numérica de las biomoléculas

La idea detrás de la caracterización numérica de secuencias de ADN y proteínas es proveer descriptores matemáticos que permitan capturar la esencia de la composición y distribución de las bases que componen las secuencias de una manera cuantitativa, de modo que facilite la identificación y comparación de similitudes y disimilitudes de secuencias. La composición de una secuencia se refiere al contenido total de cada una de las bases que integran la secuencia, siendo su determinación fácil de calcular. La distribución de una secuencia es más informativa y capaz de diferenciar entre varios genes y proteínas, aún cuando se presente igual composición en las secuencias comparadas.

Con los valores obtenidos de la caracterización numérica se espera que proporcionen cercanas homologías cuando proteínas y genes que compartan relaciones evolutivas, sean comparados con otros genes y proteínas de especies de su familia; mientras que para otros genes y proteínas los valores obtenidos sean bastantes diferentes. La composición y características de la distribución de una secuencia formarían parte de un conjunto de descriptores con los cuales se cuantificaría cada secuencia de gen o proteína.

El objetivo de los métodos de caracterización numérica para secuencias de ADN y proteínas propuestos por varios autores es formular un número que describa la distribución de una secuencia. Existen dos enfoques para definir dichos descriptores: geométricos y gráfico teóricos (Nandy et al., 2006). En este capítulo se abordan los principales elementos de estos enfoques, y se particularizará en el cálculo de los índices topológicos y los *momentos espectrales*.

2.1 Enfoque geométrico

El enfoque geométrico, realizado por primera vez por Raychaudhury y Nandy (Raychaudhury and Nandy, 1999) es derivado de las representaciones gráficas de secuencias de ADN y proteínas en cuadrículas rectangulares 2D, que empleaban un par coordenado (*x, y*) para representar numéricamente cada base de la secuencia. Dada las representaciones cartesianas de las secuencias de ADN y proteínas se definían los momentos (μ_x, μ_y) y el radio del grafo (g_R) mediante las fórmulas:

$$\mu_x = \frac{\sum x_i}{N} \quad , \quad \mu_y = \frac{\sum y_i}{N} \quad , \quad g_R = \sqrt{{\mu_x}^2 + {\mu_y}^2}$$

donde los (x_i , y_i) representan las coordenadas de cada punto de la gráfica y *N* es el número total de bases en el segmento. Aquí el (g_R) representa el índice de distribución de cada base y depende de la posición de cada base en la secuencia. La definición de (g_R) y los momentos de primer orden (μ_x, μ_y) también permiten calcular similitudes y disimilitudes entre los grafos que describen las secuencias por medio de los índices definidos como:

$$\Delta g_R = \sqrt{(\mu_{1x} - \mu_{2x})^2 + (\mu_{1y} - \mu_{1y})^2}$$

donde μ_1 y μ_2 se refieren a dos secuencias diferentes de ADN o proteínas. Los índices g_R y Δg_R han mostrado ser dos medidas muy sensitivas de la composición y distribución de secuencias. Los valores que ellos proporcionan dependen del tipo de mutación ocurrida en una secuencia así como del lugar en que esta ocurre. Δg_R es especialmente útil comparando secuencias de igual longitud.

2.2 Enfoque grafo-teórico

En el enfoque grafo teórico, una secuencia de ADN o proteína se representa por un grafo *G* = (*V*, *E*), donde *V* es un conjunto no vacío de vértices que representan los monómeros por los cuales están constituidas las mismas (nucleótidos en el caso del ADN y aminoácidos para las proteínas) y *E* es el conjunto de aristas. Tales grafos pueden ser representados como matrices de adyacencia.

Para este enfoque una distancia grafo teórica *D* puede ser definida como D_{ij} = (d_{ij}), donde d_{ij} es la cantidad de aristas entre los vértices *i* y *j*, es decir, la longitud del camino entre los vértices *i* y *j*. Varias han sido las invariantes grafo teóricas que han sido formuladas empleando distintos tipos de matrices. Una particular, es la matriz *D*/*D*, donde los valores propios de la misma han sido utilizados para cuantificar la forma de estos grafos. Los elementos de la matriz D_E/D_G son $(d_E/d_G)_{ij}$, donde d_E representa la distancia Euclidiana entre los vértices *i* y *j*; y donde d_G es la distancia grafo teórica referida anteriormente entre el par de vértices (*i, j*). El empleo de estas matrices se justifica por el hecho de que el vector formado por los valores propios de la misma es una medida del grado de doblez o plegado de la estructura que el grafo representa, de ahí el interés de esta matriz en la química estructural. El uso de esta metodología ha sido considerado como un buen descriptor para secuencias de genes y proteínas (Randic et al., 2000).

Para comparar dos secuencias se obtienen los vectores formados por los valores propios de sus respectivas matrices *D*/*D*. Luego se halla la distancia entre los mismos. Dos secuencias se pueden considerar similares si el valor obtenido de la distancia entre ambos vectores es pequeño. Otro

enfoque de comparación utiliza el ángulo que existe entre ambos vectores. Dos secuencias son muy similares si el ángulo entre los dos vectores esta cercano a cero y relativamente disímiles en otro caso.

Para ilustrar la construcción de la matriz *D*/*D* se emplea el siguiente segmento de ADN ATGGTGCACCTG formado por 12 bases. El primer paso consiste en graficar dicha secuencia en un mapa cartesiano, se utiliza para ello la representación Nandy, Figura 2.1. Al diagrama de la figura se le han añadido las etiquetas 1 – 12 para indicar la posición de cada base en el gráfico.

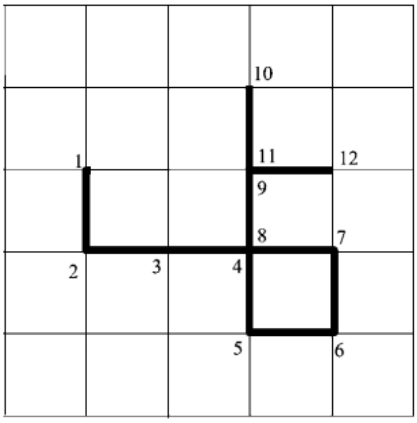


Figura 2.1. Representación cartesiana tipo Nandy de la secuencia ATGGTGCACCTG con los pasos enumerados.

La matriz D/D asociada a este grafo se muestra en la Figura 2.2. Como se puede observar, todos los elementos de la matriz para las bases que son adyacentes son iguales a uno, porque para ellos la distancia Euclidiana y la distancia grafo teórica son iguales. Para los restantes elementos de la matriz, se evalúa la distancia Euclidiana entre ellos y se divide por la distancia a lo largo del camino, que está dada por el número de aristas que los separan (en vista de que todos los pasos son de la misma longitud). Por ejemplo, el elemento (2,5) es igual a $\sqrt{5}/3$, pues la distancia Euclidiana que los separa se calcula por Pitágoras $\sqrt{2^2 + 1^2}$ = $\sqrt{5}$, mientras que el número de aristas que los separan a lo largo del camino es tres. Observar también que, aunque la distancia Euclidiana entre la primera y cuarta base, y la primera y la octava base es la misma (igual a $\sqrt{5}$), los elementos de matriz (1, 4) y (1,8) son diferentes, pues el número de pasos que separan las bases 1 y 3, y 1 y 8 son diferentes, siendo tres y siete enlaces, respectivamente.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	√2/2	√5/3	√8/4	√13/5	√10/6	√5/7	2/8	√5/9	2/10	3/11
2		0	1	2/2	√5/3	√10/4	3/5	2/6	√5/7	√8/8	√5/9	√10/10
3			0	1	√2/2	√5/3	2/4	1/5	√2/6	√5/7	√2/8	√5/9
4				0	1	√2/2	1/3	0	1/5	2/6	1/7	√2/8
5					0	1	√2/2	1/3	2/4	3/5	2/6	√5/7
6						0	1	√2/2	√5/3	√10/4	√5/5	2/6
7							0	1	√2/2	√5/3	√2/4	1/5
8								0	1	2/2	1/3	√2/4
9									0	1	0	1/3
10										0	1	√2/2
11											0	1
12												0

Figura 2.2. Matriz *D*/*D* asociada a la representación grafica de la Figura 2.1

2.3 Índices topológicos

Los índices topológicos (IT) son otra forma grafo teórica de caracterizar numéricamente representaciones gráficas de secuencias de ADN y proteínas. Los valores numéricos que ellos proporcionan a diferencia de los obtenidos de las matrices *D*/*D* no se basan en conceptos de distancias, sino en codificaciones que se hacen de la estructura de las moléculas a través de información química que se le es incluida a los grafos que las representan. Para ello el grafo molecular se convierte en un grafo valorado, donde sus nodos y aristas son ponderados con propiedades físico-químicas de los monómeros que simbolizan (Estrada, 1998).

La idea que se persigue con la representación de las biomoléculas mediante grafos y su posterior caracterización numérica a través de los índices topológicos es poder plegar las secuencias primarias de genes y proteínas en estructuras secundarias artificiales que muestren características relevantes de las reales como son hidrofobicidad, polaridad, solubilidad, composición, entre otras.

Los *momentos espectrales* (μ_k) previamente introducidos por Estrada (Estrada, 1996, Estrada, 1997) son un ejemplo de índices topológicos. El *momento espectral* de orden *k* es definido como la traza o suma de los elementos de la diagonal principal de la matriz de adyacencia de enlace correspondiente a un grafo molecular elevada al grado *k*. Esta es una matriz cuadrada simétrica cuyos elementos no diagonales son unos o ceros si los enlaces correspondientes comparten un átomo o no. Los elementos de la diagonal principal se ponderan con los pesos de enlace que describen propiedades de hidrofobicidad/polaridad, electrónicas y estéricas de las moléculas. Pueden ser también contribuciones de enlaces a las propiedades fisicoquímicas como: coeficiente de partición, área de

superficie polar, polarizabilidad, cargas atómicas Gasteiger Marsilli, radio atómico de Van der Waals y refracción molar (Brown, 1999).

En el trabajo inicial de Estrada (Estrada, 1996), se definieron los *momentos espectrales* para moléculas pequeñas, sin embargo, en investigaciones posteriores que se han hecho como continuación de su trabajo en el Centro de Bioactivos Químicos (CBQ) de la Universidad Central Marta Abreu de las Villas, se ha extendido la aplicación de los *momentos espectrales* a genes y proteínas (Agüero-Chapin et al., 2010, Agüero-Chapin et al., 2011a, Agüero-Chapin et al., 2011b).

2.4 Cálculo de los momentos espectrales

Para caracterizar una secuencia de gen o proteína mediante los *momentos espectrales*, se hace necesario codificar su secuencia primaria en un grafo. Para ello se emplean las representaciones gráficas que se hacen de estas biomoléculas vistas anteriormente. A continuación se ejemplifica el uso de los *momentos espectrales* empleando representaciones cartesianas de tipo Nandy de genes y proteínas, así como mapas de colores.

En la Figura 2.3 (a) se muestra la representación cartesiana de la secuencia (D_1-E_2-D_3-K_4-V_5). Cada uno de los nodos del grafo se ha etiquetado con los aminoácidos que simbolizan dado el mapeo de la secuencia que se ha hecho en el plano siguiendo la distribución de la Figura 1.7 y la clasificación de los aminoácidos en polares, no polares, básicos y ácidos de la Tabla 1.3. Notar que el nodo central contiene a E y a K. La matriz de adyacencia de enlace *B* es una matriz donde cada una de sus filas y columnas se hace corresponder con una arista del grafo siendo:

$$B_{ij} = \begin{cases} 1 & \text{si la arista } i \text{ y la arista } j \text{ presentan un vértice en común} \\ 0 & \text{en otro caso} \end{cases}$$

Dicha matriz es una matriz cuadrada simétrica. Para incluir información química en la representación hecha de la proteína los elementos de la diagonal principal se ponderan con pesos de enlace que describen propiedades de hidrofobicidad/polaridad, electrónicas y estéricas de los aminoácidos. Particularmente para el ejemplo desarrollado se ponderará el elemento B_{ii} con el promedio de la carga electrostática (*Q*) entre los dos nodos enlazados pertenecientes a la arista *i*, que a su vez fueron ponderados con carga electrostática *(q)* del campo de fuerza Amber 95. "*q*" es igual a la suma de las cargas de todos los aminoácidos ubicados en el nodo. La matriz de adyacencia de enlace del grafo que representa el fragmento de secuencia analizada se muestra en la Figura 2.3 (b).

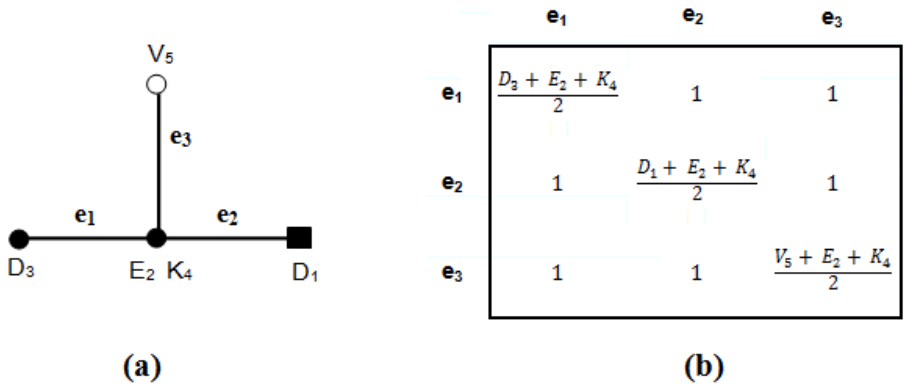


Figura 2.3. (a) Representación cartesiana del fragmento proteico DEDKV. (b) Matriz de adyacencia de enlace asociada a la representación anterior.

Luego los *momentos espectrales* de orden *k* se expresan mediante la fórmula:

$$\mu_k = Tr[(B)^k] \tag{2.1}$$

donde *Tr* es el operador traza que indica la suma de todos los valores en la diagonal principal de la matriz.

A continuación se calculan los momentos espectrales de orden uno, dos y tres. Los valores de carga electrostática del campo de fuerza Amber 95 para los aminoácidos D, E, K, V son 1.997, 1.885, 2.254 y 2.24 respectivamente. El valor de μ_0 coincide con el número de filas de la matriz *B*, pues $(B)^0$ es igual a la matriz identidad.

$$\mu_1 = Tr[(B)^1] = Tr\left[\begin{pmatrix} 3.068 & 1 & 1 \\ 1 & 3.068 & 1 \\ 1 & 1 & 3.189 \end{pmatrix}\right] = 9.325$$

$$\mu_2 = Tr[(B)^2] = Tr\left[\begin{pmatrix} 3.068 & 1 & 1 \\ 1 & 3.068 & 1 \\ 1 & 1 & 3.189 \end{pmatrix}^2\right] = Tr\left[\begin{pmatrix} 11.412 & 7.136 & 7.257 \\ 7.136 & 11.412 & 7.257 \\ 7.257 & 7.257 & 12.169 \end{pmatrix}\right] = 34.995$$

$$\mu_3 = Tr[(B)^3] = Tr\left[\begin{pmatrix} 3.068 & 1 & 1 \\ 1 & 3.068 & 1 \\ 1 & 1 & 3.189 \end{pmatrix}^3\right] = Tr\left[\begin{pmatrix} 49.406 & 40.562 & 41.691 \\ 40.562 & 49.406 & 41.691 \\ 41.691 & 41.691 & 53.323 \end{pmatrix}\right] = 152.137$$

Para la representación de proteínas en un mapa de cuatro colores se considera un nodo del grafo una agrupación de aminoácidos del mapa o región. Dos regiones son adyacentes si presentan un borde en común. Para el cálculo de los momentos espectrales en representaciones de mapas de colores de genes y proteínas la matriz *B* que se emplea es la matriz de adyacencia de nodos, a diferencia de la matriz de adyacencia de enlace que era la utilizada en las representaciones cartesianas. Debido a esto el número de nodos del grafo o de regiones del mapa será igual al número de filas y columnas de la matriz *B*. Como una región del mapa estará formada por un conjunto de aminoácidos que comparten propiedades fisicoquímicas similares, el nodo en el grafo que representa esa región se ponderará como la suma de las propiedades individuales (*q*) de cada aminoácido que integra el nodo. La suma de las propiedades fisicoquímicas que pondera un nodo se denomina *Q*.

Luego, la matriz *B* derivada de representaciones como mapas de colores de genes y proteínas se define como:

$$B_{ij} = \begin{cases} (Q_i + Q_j)/2 & \text{si el nodo } i \text{ y } j \text{ son adyacentes y si } i \neq j \\ 0 & \text{si los nodos } i \text{ y } j \text{ no son adyacentes y si } i \neq j \\ Q_i & \text{si } i = j \end{cases} \quad (2.2)$$

Como ejemplo, se emplea el segmento de secuencia (M_1-V_2-N_3-S_4-S_5-K_6-S_7-I_8-L_9). El mapa de cuatro colores y la matriz *B* asociada a esta secuencia de aminoácidos se puede apreciar en la Figura 2.4.

Los valores (*q*) asociados a cada aminoácido serán la carga electrostática del campo de fuerza Amber 95 (M=1.91, V= 2.24, N=2.07, S=2.09, K= 2.25, I= 2.03, L= 1.91). Los *momentos espectrales* para *k* = 0, 1, 2 de la secuencia analizada se calculan según la expresión (2.1).

$$\mu_0 = Tr[(B)^0] = Tr\left[\begin{pmatrix} 8.09 & 7.17 & 5.09 & 5.17 \\ 7.17 & 6.25 & 0 & 4.25 \\ 5.09 & 0 & 2.09 & 2.17 \\ 5.17 & 4.25 & 2.17 & 2.25 \end{pmatrix}^0\right] = 4$$

$$\mu_1 = Tr[(B)^1] = Tr\left[\begin{pmatrix} 8.09 & 7.17 & 5.09 & 5.17 \\ 7.17 & 6.25 & 0 & 4.25 \\ 5.09 & 0 & 2.09 & 2.17 \\ 5.17 & 4.25 & 2.17 & 2.25 \end{pmatrix}^1\right] = 18.68$$

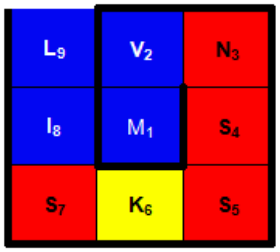
$$\mu_2 = Tr[(B)^2] = Tr\left[\begin{pmatrix} 8.09 & 7.17 & 5.09 & 5.17 \\ 7.17 & 6.25 & 0 & 4.25 \\ 5.09 & 0 & 2.09 & 2.17 \\ 5.17 & 4.25 & 2.17 & 2.25 \end{pmatrix}^2\right]$$

$$= Tr\left[\begin{pmatrix} 169.494 & 124.790 & 63.035 & 94.976 \\ 124.790 & 108.534 & 45.718 & 73.194 \\ 63.035 & 45.718 & 34.895 & 35.733 \\ 94.976 & 73.194 & 35.733 & 54.563 \end{pmatrix}\right] = 367.486$$

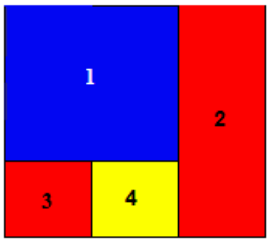
(a)

$M_1 - V_2 - N_3 - S_4 - S_5 - K_6 - S_7 - I_8 - L_9$

(b)



(c)



(d)

	1	**2**	**3**	**4**
1	$L_9 + V_2 + M_1 + I_8$	$\frac{L_9 + V_2 + M_1 + I_8 + N_3 + S_4 + S_5}{2}$	$\frac{L_9 + V_2 + M_1 + I_8 + S_5}{2}$	$\frac{L_9 + V_2 + M_1 + I_8 + K_6}{2}$
2	$\frac{L_9 + V_2 + M_1 + I_8 + N_3 + S_4 + S_5}{2}$	$N_3 + S_4 + S_5$	0	$\frac{N_3 + S_4 + S_5 + K_6}{2}$
3	$\frac{L_9 + V_2 + M_1 + I_8 + S_7}{2}$	0	S_7	$\frac{K_6 + S_7}{2}$
4	$\frac{L_9 + V_2 + M_1 + I_8 + K_6}{2}$	$\frac{N_3 + S_4 + S_5 + K_6}{2}$	$\frac{K_6 + S_7}{2}$	K_6

Figura 2.4. (a) Secuencia de aminoácidos. (b) La secuencia plegada dentro de una cuadrícula en forma de espiral (c) Representación en mapa de colores de la secuencia. (d) Matriz *B* asociada al mapa de cuatro colores.

2.5 Consideraciones finales del capítulo

Al calcular ciertas propiedades topológicas a partir de los grafos utilizados para representar biomoléculas, se puede establecer una relación entre la estructura de la molécula y su actividad. Estos modelos son libres de alineamientos, o sea, la evaluación de las similitudes en las cadenas de

ARN, ADN y proteínas no se basan en la comparación lineal de las secuencias de bases nitrogenadas que lo conforman, sino en el enfoque grafo-teórico de la representación de la molécula. De esta forma se pueden proveer descriptores matemáticos que permiten capturar la esencia de la composición y distribución de las bases que componen las secuencias de una manera cuantitativa, de modo que facilite la identificación y comparación de similitudes y disimilitudes de secuencias. Una vertiente con éxito en tal sentido es el cálculo de los índices topológicos a partir de grafos, particularmente los *momentos espectrales*.

3 ESPECTRO: software que permite representar, describir y caracterizar secuencias de ADN y proteínas

En este capítulo se exponen las principales características del software ESPECTRO, desarrollado como resultado de esta investigación para facilitar la caracterización numérica de secuencias de ADN y proteínas a partir de sus representaciones gráficas. Se especificará el actor del sistema, sus casos de uso y el diagrama de clases, detallando las principales clases y sus interacciones en el sistema general.

3.1 Generalidades de ESPECTRO

En este trabajo se desarrolló el software ESPECTRO con el objetivo de brindar una herramienta a los investigadores en Bioinformática que permita caracterizar numéricamente secuencias de ADN y proteínas a partir de sus representaciones gráficas. En el software se implementa una metodología libre de alineamiento que se basa en la teoría de grafos y que permite extraer información de dichas biomoléculas. La metodología empleada es una extensión de la definida en (Estrada, 1996, Estrada, 1997) para el cálculo de los *momentos espectrales*. El cálculo de los *momentos espectrales* se usa para llevar a cabo la Relación Cuantitativa Secuencia Función (QSFR) la cual permite la clasificación de clases de genes y proteínas sin necesidad de ejecutar un procedimiento de alineamiento.

ESPECTRO 1.0 también es una herramienta que permite obtener representaciones gráficas 2D de genes y proteínas, posibilitando realizar inspecciones visuales de fragmentos de secuencias de ADN y proteínas en búsqueda de similitudes y disimilitudes. Las formas de representación gráficas implementadas en (ESPECTRO *versión 1.0 ®)* son la cartesiana de Nandy (Nandy, 1994)y la de mapas de colores (Randic and Balaban, 2005). Se emplean éstas pues son las que más se reportan en la literatura, aunque en el epígrafe 1.3 se vieron otras existentes que también resultan de utilidad y se pudieran incorporar fácilmente.

El diseño realizado del software es extensible; y posibilita que alternativas distintas de ponderación de la matriz que describe el grafo molecular puedan ser empleadas. Por tanto, el código queda abierto para el empleo futuro de nuevas alternativas para el cálculo de los momentos espectrales, así como para la inclusión de nuevas formas de representación gráfica 2D de genes y proteínas.

3.2 Plataforma de desarrollo

Se escogió el lenguaje Java para la implementación de (ESPECTRO *versión 1.0* ®*).* Java es un lenguaje de programación que es distribuido actualmente como software libre bajo una licencia GNU GPL (General Public License), lo cual lo hace un gran candidato para el uso en el desarrollo de aplicaciones en países del tercer mundo. El lenguaje Java fue creado para trabajar con objetos y de manera independiente a la plataforma. Esto es posible debido a la JVM (Java Virtual Machine) la cual es una máquina genérica, que ejecuta un código creado al compilar un programa, el cual corre indistintamente en cualquier ordenador que tenga instalada dicha máquina virtual. Java es un lenguaje robusto justamente por la forma en que está diseñado, no permite el manejo directo del hardware ni de la memoria. Implementa, además, mecanismos de seguridad que limitan el acceso a recursos de las máquinas donde se ejecuta.

Para el trabajo con gráficos se empleó *G*, una biblioteca genérica de código abierto (open source) desarrollada sobre Java 2D y que se distribuye bajo licencia (LGPL). Entre sus características destacan: orientada a objetos gráficos de escena jerárquicos, gráficos en capas con soporte de visibilidad, potentes funciones de detección de objetos, soporte para trabajo en coordenadas de mundo y de dispositivo, soporte para interacciones con el usuario, utilidades para el trabajo geométrico y de transformación de coordenadas, peso ligero (aproximadamente 80 KB), autónoma y fácil de usar.

Se implementó el análisis de los datos y la ejecución del algoritmo usando programación multitarea basada en hilos. La programación multitarea permite la realización concurrente de varias actividades en la computadora. Este tipo de programación puede dividirse en multitarea basada en hilos o basada en procesos. La programación multitarea basada en hilos permite ejecutar partes de un programa concurrentemente, la secuencia de código ejecutado por cada tarea define caminos separados de ejecución a los que se llaman hilos. Como ventajas de la programación multitareas basada en hilos se pueden mencionar (Mughal and Rasmussen, 2004):

- buen uso del tiempo en que la CPU está desocupada,
- los hilos comparten el mismo espacio de memoria, y
- el costo de comunicación entre ellos es relativamente bajo.

Java soporta multitarea basada en hilos y provee facilidades para la programación multihilos.

Entre los IDE (Integrated Development Environment) disponibles para Java se seleccionó el NetBeans 6.5. Es un ambiente de programación cómodo, que compila en tiempo real y fácil de usar para depurar un programa.

3.3 Análisis y diseño del software

Para realizar el análisis y diseño de la herramienta (ESPECTRO *versión 1.0 ®)*, se utilizaron las facilidades del lenguaje UML (Unified Modeling Language) (Rumbaugh and Booch, 2000) que tiene como objetivos principales la especificación, visualización, construcción y documentación de los productos de un sistema de software. Este lenguaje es usado por el RUP (Rational Unified Process) (Jacobson et al., 2000) como lenguaje de modelado para lo cual se basa en todos sus tipos de diagramas, que constituyen diferentes vistas del modelo del producto. La Figura 3.1 ilustra los diagramas que componen la estructura de un producto escrito por el lenguaje UML.

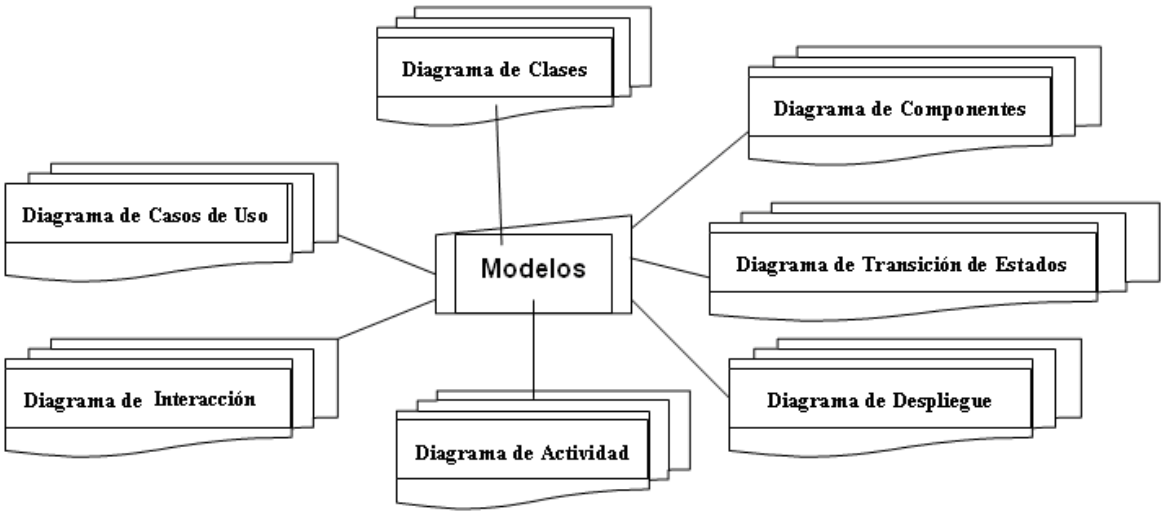


Figura 3.1. Diagrama de UML.

De los diagramas de UML que muestra la Figura 3.1, se emplearon el diagrama de casos de uso y el diagrama de clases. La herramienta empleada para el modelado de los diagramas correspondientes a las fases de análisis y diseño fue Visual Paradigm 6.0 Enterprise Edition.

3.3.1 Casos de uso

Los modelos de casos de uso proporcionan un medio sistemático e intuitivo para capturar requisitos funcionales del sistema basándose en los requerimientos de los usuarios. Ellos dirigen todo el proceso de desarrollo de un software ya que constituyen el punto de partida para llevar a cabo la mayoría de las actividades: el análisis, diseño y prueba del software (Jacobson et al., 2000). Este modelo se realiza identificando cada actor del sistema como los posibles usuarios para los cuales está realizado el mismo.

La herramienta ESPECTRO está destinada a un solo tipo de actor que es el especialista o investigador sobre el área de la Bioinformática que desee realizar estudios comparativos de genes y proteínas y fundamentalmente para el interesado en caracterizar numéricamente secuencias de ADN y proteínas para la posterior construcción de modelos de clasificación. Este tipo de especialista debe ser capaz de interpretar de forma correcta los resultados que brinda el software. Dicho especialista se ha nombrado usuario en el diagrama de la Figura 3.1

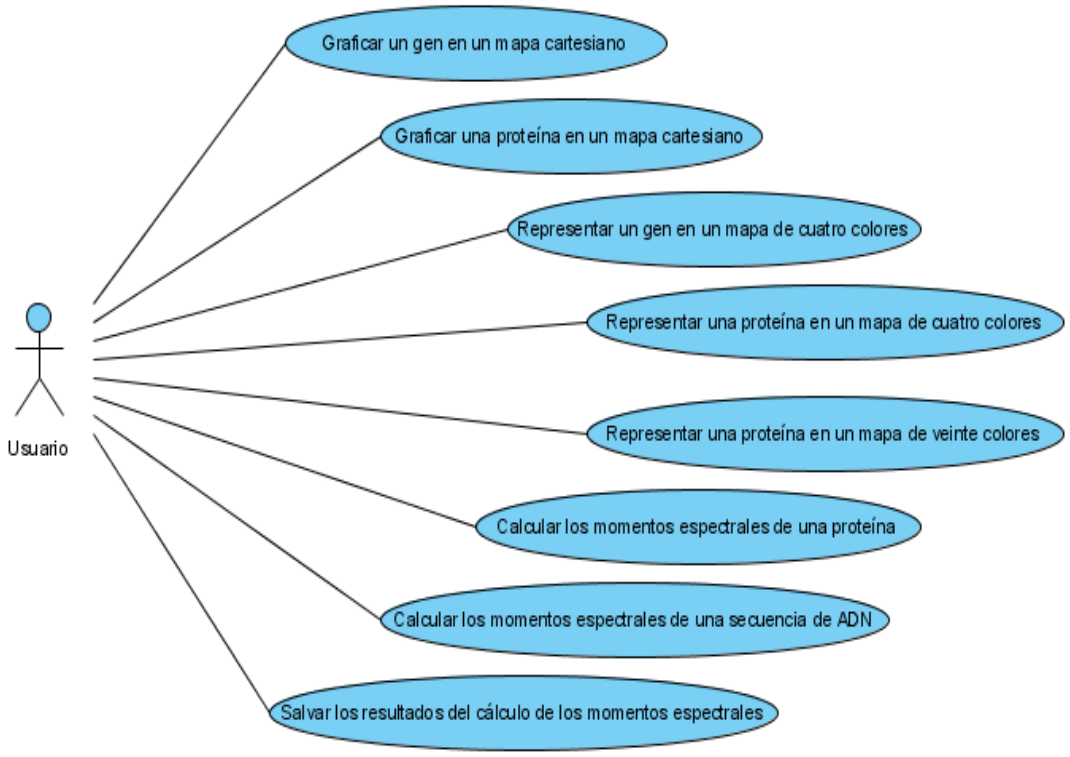


Figura 3.1. Diagrama de casos de usos

En la Tabla 3.1 se describen los casos de uso del actor usuario.

Tabla 3.1. Descripción de los casos de uso.

Casos de uso	Descripción
Cargar datos	Permite seleccionar el archivo que contiene la descripción de las secuencias de ADN o proteína (*.fasta, *.gb, *.gp, *.pdb)
Graficar un gen en un mapa cartesiano	Permite graficar una secuencia de ADN en un sistema cartesiano de dos dimensiones. Las secuencias de nucleótidos son ploteadas en un camino a través de una cuadrícula, usando cuatro direcciones cardinales representadas cada una por una base.
Graficar una proteína en un mapa cartesiano	Permite graficar una secuencia de proteína en un sistema cartesiano de dos dimensiones. Las secuencias de aminoácidos son ploteadas en un camino a través de una cuadrícula.
Representar un gen en un mapa de cuatro colores	Permite obtener una imagen de un gen que ha sido representado gráficamente en un mapa de cuatro colores. Una secuencia de ADN se pliega en forma de espiral dentro de una cuadrícula y se le asigna un color diferente a cada base. La imagen queda formada por regiones coloreadas que determinan agrupaciones de nucleótidos.
Representar una proteína en un mapa de cuatro colores.	Permite obtener una imagen de una proteína que ha sido representada gráficamente en un mapa de cuatro colores. La imagen queda formada por regiones coloreadas que determinan agrupaciones de aminoácidos con propiedades fisicoquímicas comunes.
Representar una proteína en un mapa de veinte colores.	Permite obtener una imagen de una proteína que ha sido representada gráficamente en un mapa de veinte colores. La imagen queda formada por regiones coloreadas que determinan agrupaciones de aminoácidos de un tipo específico.

Calcular los momentos espectrales de una proteína.	El software calcula los valores de los momentos espectrales de una proteína, siendo previamente especificada por el usuario la representación gráfica que se hará de la proteína y la ponderación a emplear para cada uno de sus aminoácidos.
Calcular los momentos espectrales de una secuencia de ADN	El software calcula los valores de los momentos espectrales de una secuencia de ADN, dada la representación gráfica especificada por el usuario que se hará de la misma.
Salvar los resultados del cálculo de los momentos espectrales	Permite al usuario salvar los valores de los momentos espectrales de las secuencias analizadas. Los resultados se pueden guardar en formato de texto plano o de Excel.

3.3.2 Diagrama de clases

La técnica del diagrama de clase se ha vuelto medular en los métodos orientados a objetos. El diagrama de clase describe los tipos de objetos que hay en un sistema y las diversas clases de relaciones estáticas (asociaciones, subtipos) que existen entre ellos. También muestran los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos (Fowler and Scott, 1997).

En la Figura 3.2 se muestran los siete paquetes de clases que se crearon al diseñar el software ESPECTRO, así como las relaciones que se establecen entre los mismos. Los paquetes de clases creados son:

- ***four_color_adn***
- ***four_color_proteina***
- ***twenty_color_proteina***
- ***pam_proteina***
- ***cartesina***
- ***util***
- ***visual***

Las clases asociadas con las estructuras de datos necesarias para manipular la representación gráfica de una secuencia de ADN o proteína se agruparon en un paquete. Tales son los casos de los paquetes ***four_color_adn***, ***four_color_proteina*** y ***twenty_color_proteina***, que gestionan las clases

asociadas a la representación del ADN como mapa de cuatro colores, la representación de una proteína como mapa de cuatro colores y la representación de una proteína como mapa de veinte colores respectivamente.

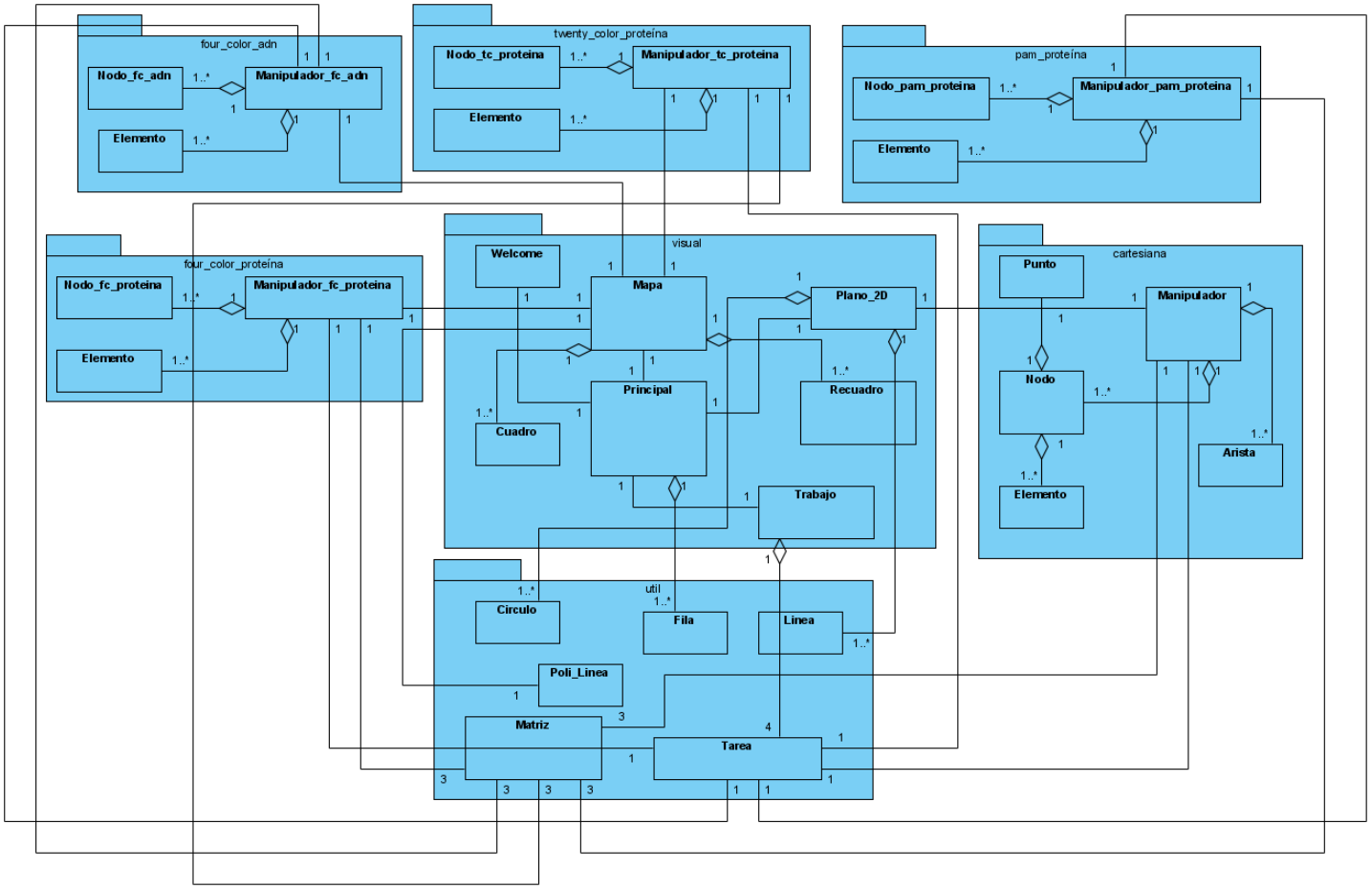


Figura 3.2. Diagrama de clases simplificado.

El paquete ***cartesiana*** gestiona las representaciones de genes y proteínas en un sistema cartesiano de dos dimensiones. El paquete ***pam_proteina*** gestiona la manipulación de una proteína cuando se pondera utilizando la matriz PAM 250. Cada uno de estos paquetes contiene una clase llamada ***Manipulador***, que es la que gestiona el contenido del paquete e interactúa con las demás clases del exterior. Dentro de cada una de estas clases ***Manipulador*** se han implementado métodos que tienen por fin simular la construcción de la representación gráfica del gen o proteína, obtener el grafo asociado a la representación gráfica hecha, obtener la matriz de adyacencia de aristas o de nodos según el caso, realizar la ponderación de la matriz anterior según lo especificado por el usuario y

calcular los valores de los momentos espectrales por solo mencionar algunos. Para la realización de estas operaciones la clase ***Manipulador*** se auxilia de la clase ***Nodo***, donde se define la estructura de un vértice del grafo y la clase ***Elemento*** que representa un monómero de la secuencia.

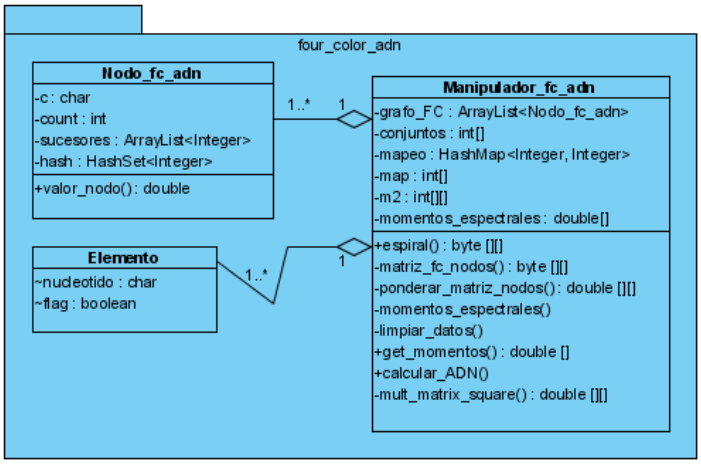


Figura 3.3. Diagrama de clases del paquete four_color_adn.

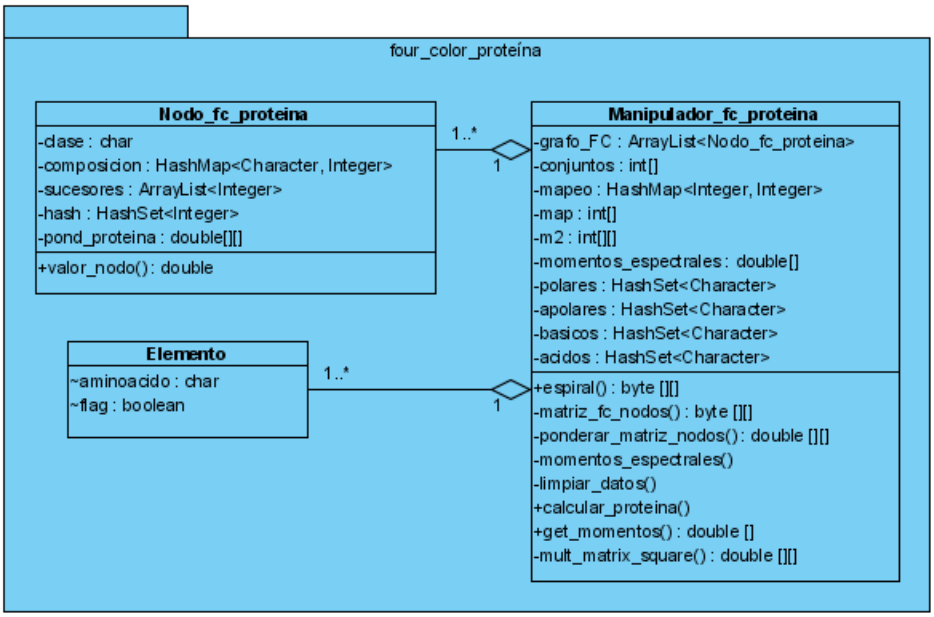


Figura 3.4. Diagrama de clases del paquete four_color_proteina.

En la Figura 3.3 y Figura 3.4 se muestran por separado los paquetes ***four_color_adn*** y ***four_color_proteina*** con la descripción de los atributos y métodos que componen cada clase. En ambas el grafo descrito por una representación en mapa de cuatro colores de una secuencia de ADN

o proteínas se implementa mediante una lista de adyacencia. El atributo ***grafo_FC*** de tipo *ArrayList<Nodo>* es la estructura que se emplea para ese fin. El método ***espiral()*** es el encargado de plegar una secuencia de gen o proteína dentro de una cuadrícula en forma de espiral, así como de construir el grafo asociado a la representación en mapa de cuatro colores de la secuencia; emplea para ello un procedimiento de búsqueda a lo ancho.

En la Figura 3.5 se muestra los atributos y métodos de las clases que integran el paquete ***cartesiana***. Dado que el cálculo de los *momentos espectrales* en representaciones cartesianas está asociado a la matriz de adyacencia de arista, se emplea una estructura de tipo *HashMap* (el atributo ***hash_arista*** de la clase ***Manipulador***) para una manipulación más eficaz de la colección de objetos tipo ***Arista*** que conforman el grafo derivado de la representación gráfica.

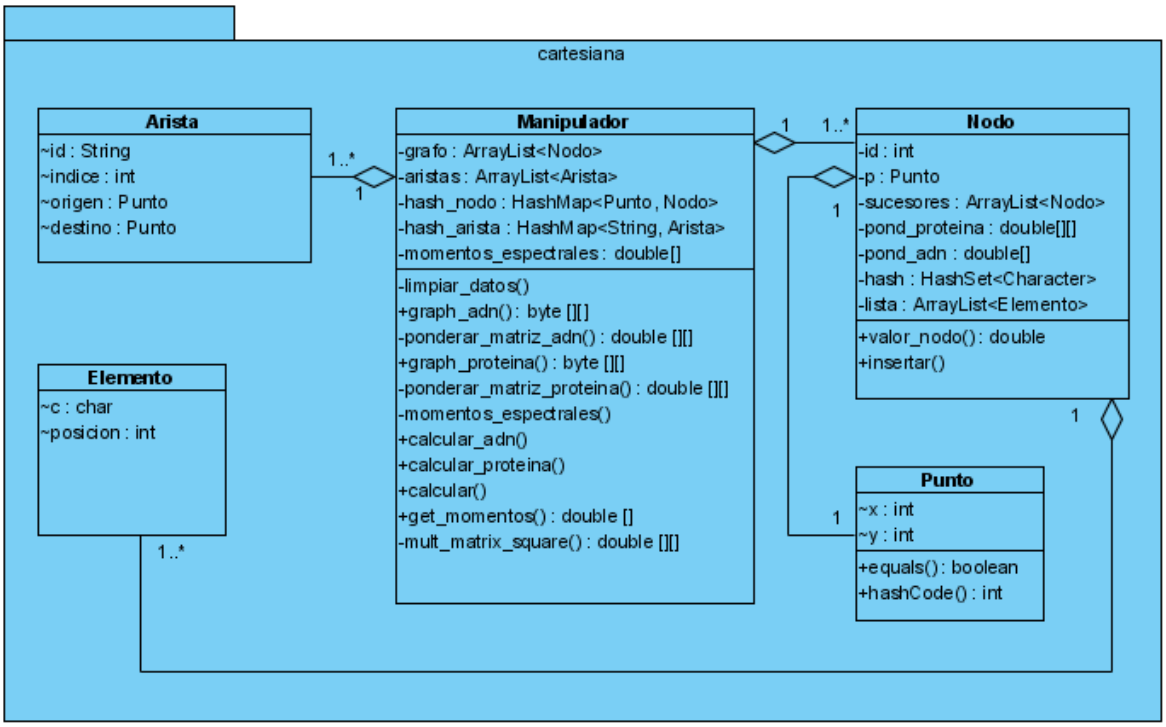


Figura 3.5. Diagrama de clases del paquete cartesiana.

En una representación cartesiana de una biomolécula, un punto del plano se hace corresponder con un nodo del grafo. El atributo ***hash_nodo*** de tipo *HashMap<Punto, Nodo>* es el que implementa esta asociación de una manera eficaz. Los métodos ***calcular_adn()*** y ***calcular_proteina()*** de la clase ***Manipulador*** son los encargados de calcular los *momentos espectrales* asociados a representaciones cartesianas de secuencias de ADN y proteínas respectivamente.

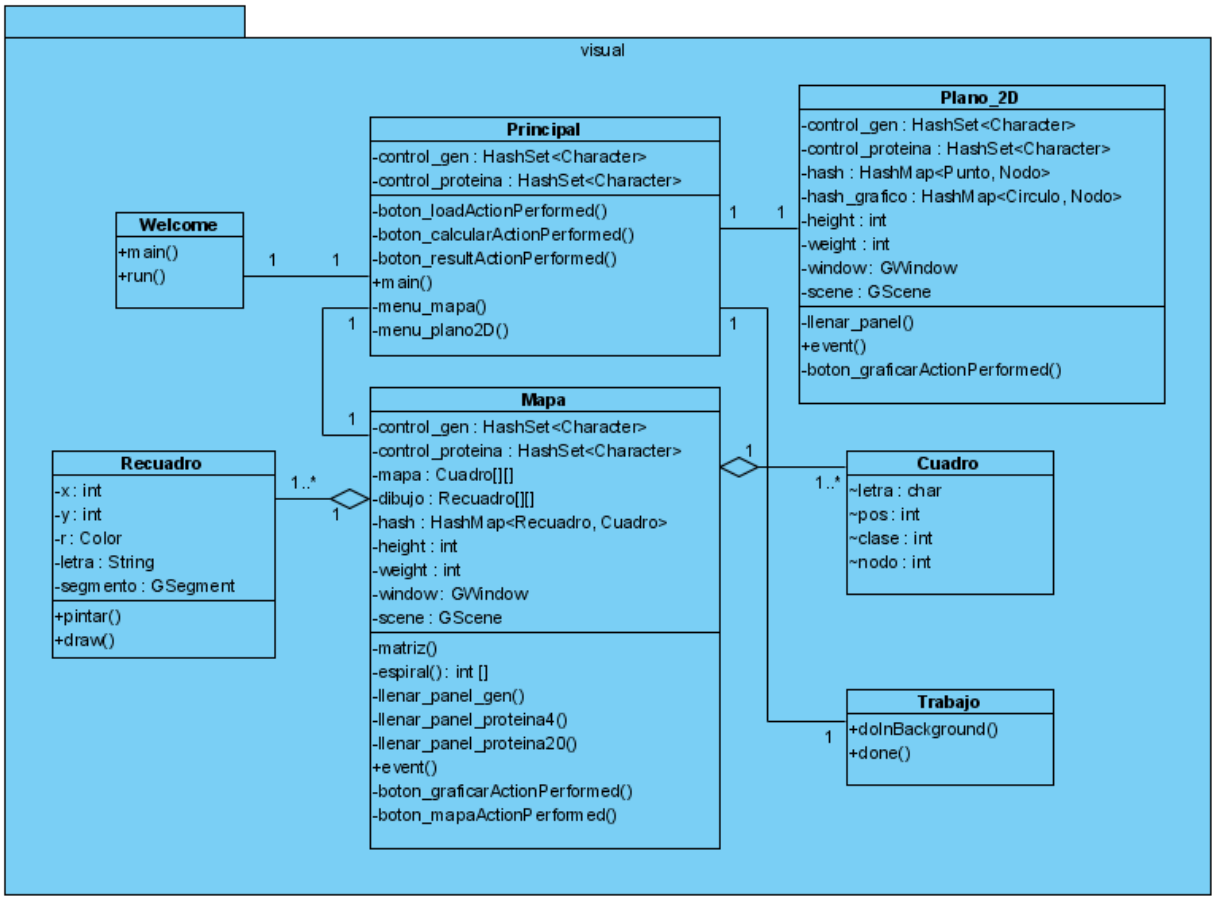


Figura 3.6. Diagrama de clases del paquete visual

El paquete ***visual*** mostrado en la Figura 3.6 incluye las clases ***Welcome***, ***Principal***, ***Mapa*** y ***Plano2D***, las cuales son las relacionadas con la interfaz visual del software. La clase ***Welcome*** contendrá una pantalla de bienvenida que será la primera mostrada al usuario. La clase ***Principal***, es el ambiente principal de trabajo. En ella es donde el usuario introduce las secuencias de genes y proteínas y procede a realizar el cálculo de los *momentos espectrales* de las mismas. La clase ***Mapa*** es la encargada de visualizar representaciones de genes y proteínas como mapa de colores. La clase ***Plano2D*** visualiza representaciones de secuencias de ADN y proteínas en un Sistema Cartesiano de dos dimensiones. Esta clase presenta un atributo ***hash_grafico*** de tipo *HashMap<Circulo, Nodo>* que asocia un objeto gráfico de tipo ***Circulo*** con un nodo del grafo. Este vínculo se hace necesario para la implementación de las interacciones que brinda el software con el usuario, en los procesos de visualización de las representaciones cartesianas. Las clases ***Cuadro***, ***Recuadro*** y ***Trabajo*** también se incluyen en el paquete ***visual***. Las dos primeras son complementarias de la clase ***Mapa*** y se emplean

para facilitar la visualización de los mapas de colores. La clase ***Trabajo*** utilizada por la clase ***Principal*** hereda de la clase abstracta *SwingWorker* incluida en la API del lenguaje Java. La misma presenta un método llamado *doInBackground()* donde se realiza el procesamiento de los datos de manera concurrente y en un hilo de ejecución de fondo independiente del proceso que ejecuta la aplicación principal.

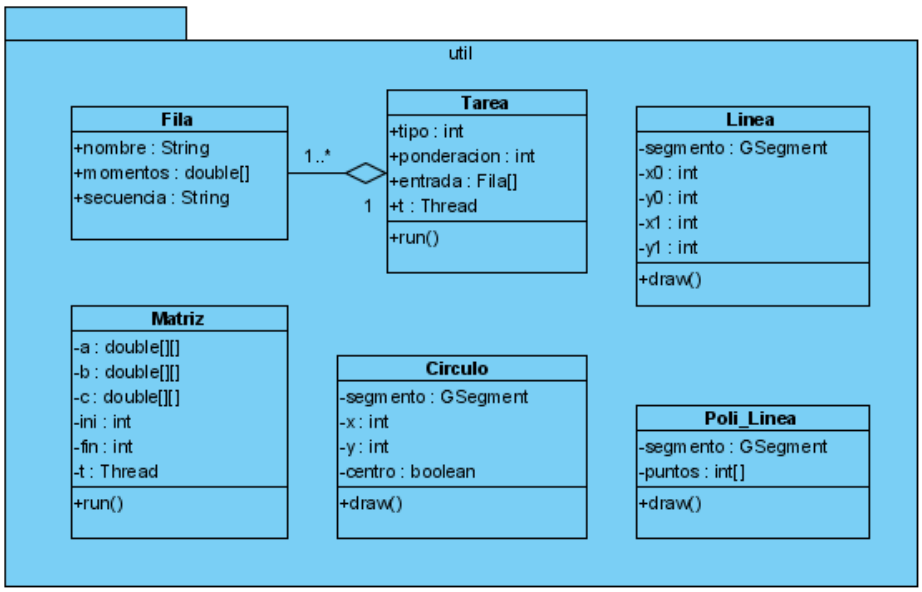


Figura 3.7. Diagrama de clases del paquete util

El paquete ***util*** mostrado en la Figura 3.7 incluye clases que complementan las funcionalidades del software y son empleadas por clases de otros paquetes. Lo componen: ***Linea***, ***Poli_Linea***, ***Circulo***, ***Fila***, ***Matriz*** y ***Tarea***. Las clases ***Linea***, ***Poli_Linea*** y ***Circulo*** heredan de la clase *GObject* incluida en la biblioteca *G*. Las instancias de estas clases son objetos gráficos que se emplean en las representaciones visuales de genes y proteínas. La clase ***Fila*** define una estructura donde se almacena la información asociada a una secuencia que se procesa.

En la clase ***Matriz*** se implementa un hilo de ejecución para la multiplicación de matrices en paralelo. El constructor de la misma recibe de parámetros las matrices «a» y «b» que se desean multiplicar, la locación de memoria de la matriz «c» donde se escribirán los resultados del proceso de multiplicación y dos parámetros enteros *inicio* y *fin* que indican la fila inicial y final de la porción de la matriz «c» que se desea obtener. Luego dentro de los métodos *mult_matrix_square()* de las clase ***Manipulador*** se crean tres instancias de la clase ***Matriz***; cada una se corresponderá con un hilo de ejecución que

calculará una porción de la matriz «c» resultado del proceso de multiplicación de las matrices «a» y «b» .

La clase ***Tarea*** define una unidad de procesamiento. Dentro del método *run()* de las mismas se crean instancias de las clases tipo ***Manipulador*** que calculan los valores de los *momentos espectrales* de las secuencias, según la representación gráfica y ponderación especificada por el usuario. Una instancia de la clase ***Tarea*** procesa un subconjunto del total de secuencias a calcular.

Básicamente cuando un usuario carga el conjunto de secuencias a procesar y presiona clic sobre el botón *Calcular* se lanza un evento que es capturado por el método *boton_calcularActionPerformed()* de la clase ***Principal.*** En este método un objeto de la clase ***Trabajo*** crea un hilo de fondo que se ejecutará concurrentemente con el de la aplicación principal. El procesamiento a ejecutar en ese hilo de fondo se especifica dentro del método *doInBackground()* de la clase ***Trabajo.*** Allí el total de secuencias cargadas por el usuario se particiona en *k* subconjuntos. Luego para cada una de las particiones se crea una instancia de la clase ***Tarea*** que se encargará del procesamiento de la misma. El resultado será *k* hilos ejecutándose concurrentemente, cada uno encargado del procesamiento de una partición. Señalar que el desempeño del software dependerá en gran medida de las prestaciones con las que contará el computador sobre el cual se corre la aplicación. La programación multihilo se hace eficiente solo cuando se cuenta con procesadores de varios núcleos.

En la Figura 3.9 y Figura 3.8 se muestran los diagramas de clases de los paquetes twenty_color_proteina y pam_proteina con los atributos y métodos de las clases que lo integran.

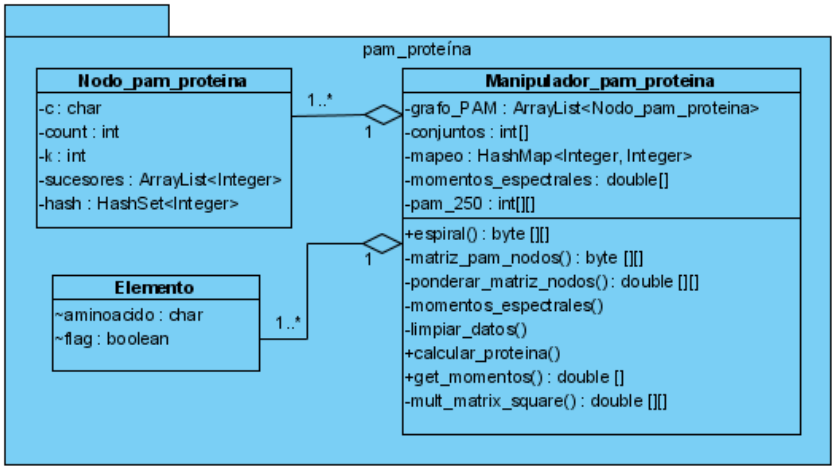


Figura 3.8. Diagrama de clases del paquete twenty_color_proteina

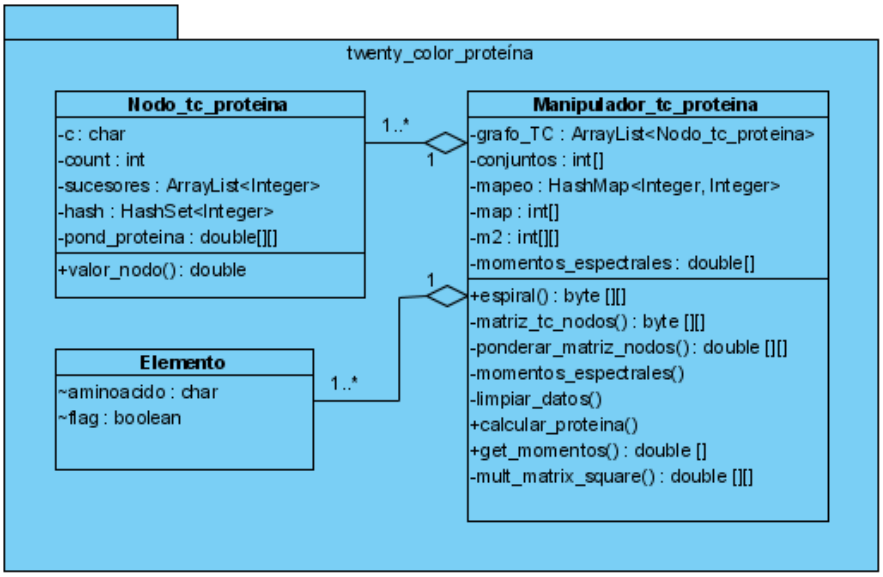


Figura 3.9. Diagrama de clases del paquete pam_proteina

3.4 Conclusiones parciales

El software ESPECTRO desarrollado permite caracterizar numéricamente secuencias de ADN y proteínas a partir de las representaciones gráficas: cartesiana de Nandy y los mapas de colores, aunque su diseño es extensible y permite incorporar otras formas de representación.

ESPECTRO permite caracterizar y comparar biomoléculas siguiendo una metodología libre de alineamiento, ya que calcula índices topológicos basados en el cálculo de los *momentos espectrales* a partir de las representaciones gráficas. Además, permite obtener representaciones gráficas 2D de genes y proteínas, posibilitando realizar inspecciones visuales de fragmentos de secuencias de ADN y proteínas en búsqueda de similitudes y disimilitudes.

En ESPECTRO, un especialista en Bioinformática, puede realizar un ciclo cerrado de análisis de las biomoléculas transitando desde cargar los datos en diferentes formatos, graficar las biomoléculas incluyendo varios tipos de representaciones, calcular los *momentos espectrales* a partir de la representación seleccionada y salvar los resultados para ser utilizados en otros sistemas que permitan predecir otros comportamientos, o construir, por ejemplo, árboles filogenéticos a partir de esa caracterización lograda.

4 ESPECTRO: funcionalidades y soporte para la predicción

En este capítulo se describen los requerimientos del software ESPECTRO, así como sus elementos principales, facilidades de uso y potencialidades para la caracterización de biomoléculas. Finalmente, se ilustran las potencialidades de las representaciones y caracterizaciones obtenidas con ESPECTRO para realizar predicciones de funciones tipo bacteriocina y en la identificación de dominios de adenilación, utilizando la herramienta Weka.

4.1 Requerimientos

Los requerimientos mínimos para la aplicación son: una computadora con 256 MB de RAM y la máquina virtual de Java (JVM) instalada, o en su defecto, el ambiente de ejecución de Java (Java Runtime Enviroment). La versión tanto de la JVM como del ambiente de ejecución de Java debe ser 1.5 o superior.

4.2 Interfaz principal

En la Figura 4.1 se muestra la interfaz principal de la aplicación. Al centro aparece una tabla de dos campos donde aparece el nombre y la secuencia de monómeros que describen una proteína o un gen introducido por el usuario. Una proteína o gen puede ser introducida al software presionando el botón *Adicionar* luego de ser previamente especificado el nombre y la secuencia que la describe en los campos de texto del panel *Biomolécula*. También el usuario puede añadir secuencias especificadas en determinado formato de un fichero presionando el botón *Cargar Fichero*. Los formatos que lee el software para secuencias de ADN pueden ser de tipo FASTA (*.fasta) o GenBank (*.gb). Para proteínas los ficheros de entrada que acepta el software son FASTA (*.fasta), GenBank (*.gp) y del ProteinDataBank (*.pdb).

Para realizar el cálculo de los *momentos espectrales* el usuario debe especificar el tipo de las secuencias que serán analizadas seleccionando una opción del panel *Procesar como*. La representación gráfica que se hará de las biomoléculas en el cálculo de los *momentos espectrales* el usuario la debe especificar en el panel *Representación*. Las opciones disponibles para secuencias de ADN, son *Cartesiana de Nandy* y *Mapa de cuatro colores.* En el caso de procesamientos de proteínas a las dos opciones anteriores se le suma la representación de *Mapa de veinte colores.* También se

hace necesario que el usuario especifique la ponderación a emplear en la matriz de adyacencia del grafo que representa la biomolécula. Por defecto aparece *sum-Amber95*, que es la carga electrostática del campo de fuerza Amber 95.

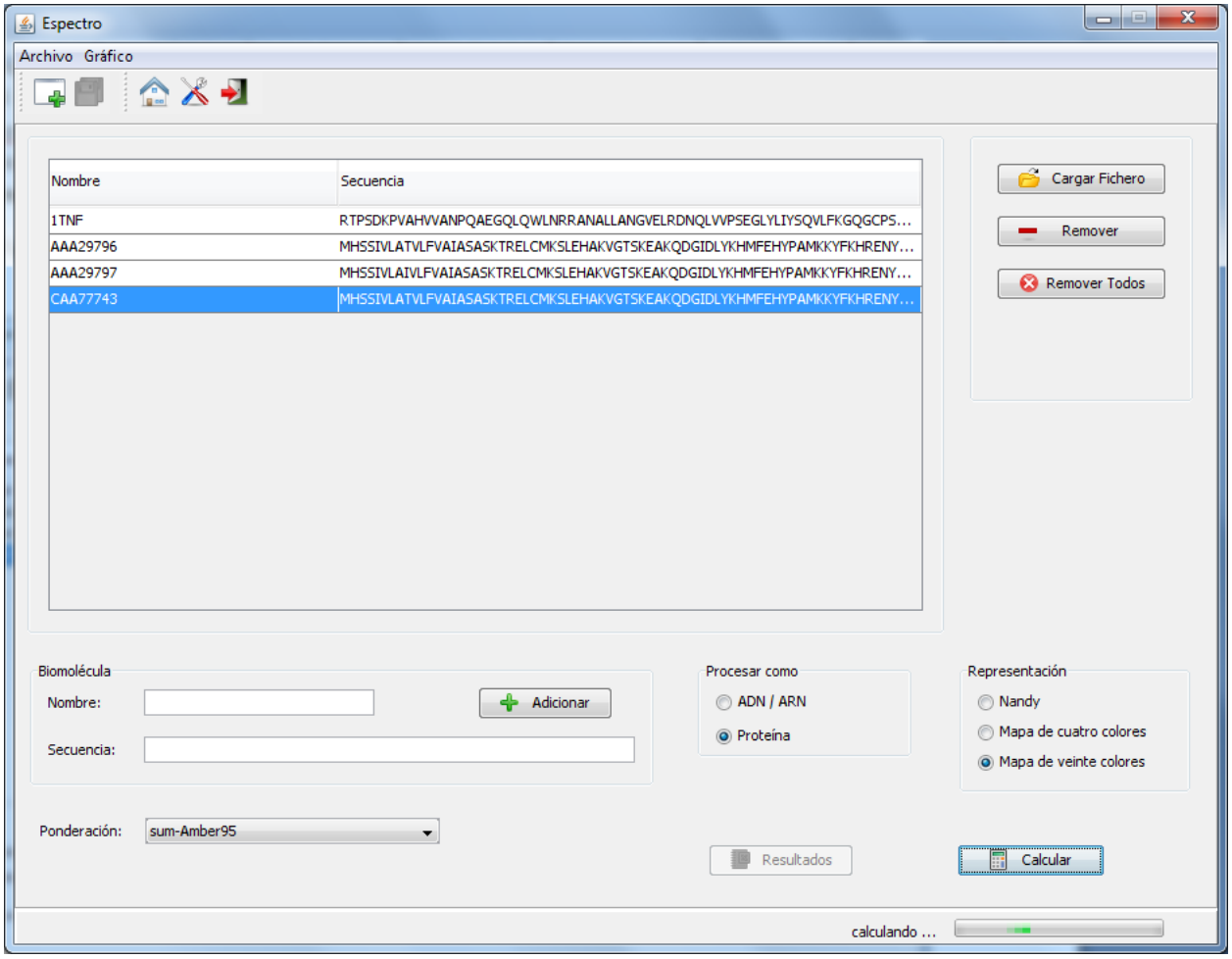
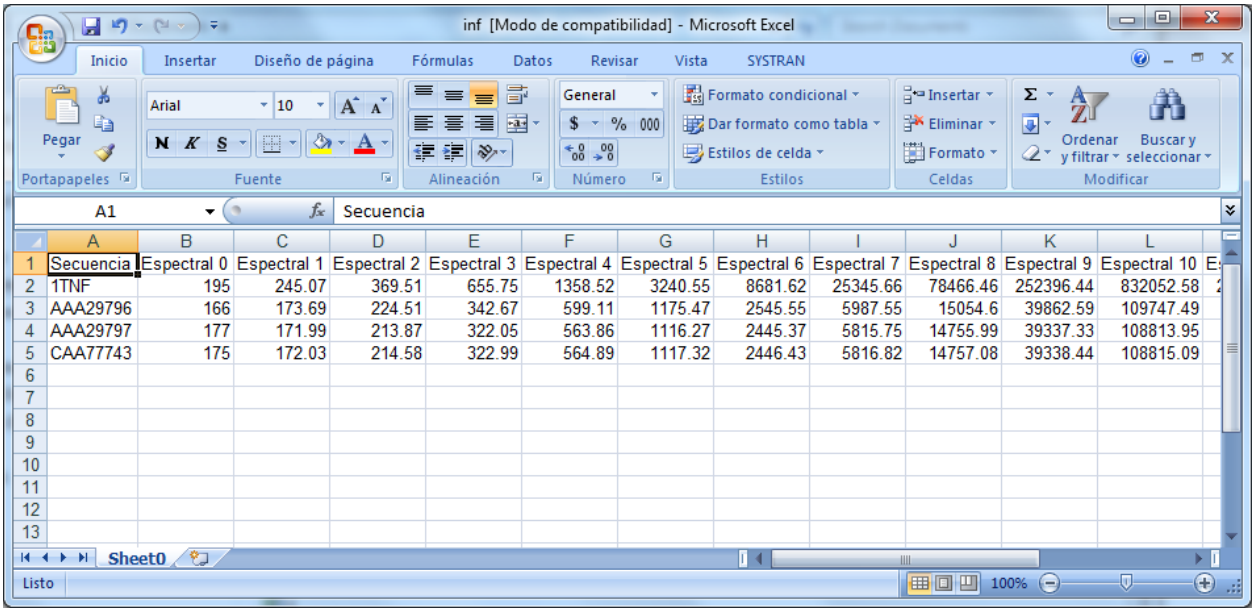


Figura 4.1. Interfaz principal de ESPECTRO *versión 1.0* ®

Una vez configurado los parámetros el usuario puede realizar el cálculo de los *momentos espectrales* de las secuencias introducidas presionando el botón *Calcular*. Para indicar al usuario que el software está ejecutando el análisis de las secuencias, se muestra en la esquina inferior derecha de la ventana principal una barra de progreso. Este procesamiento se ejecuta como tarea de fondo, por lo que no afecta el proceso de ejecución de la aplicación principal. Concluido el procesamiento se detiene la barra de progreso y se activa al botón *Resultados*, donde el usuario puede acceder para guardar los valores obtenidos del cómputo de los datos. Los resultados se pueden almacenar en formato de texto

plano o en formato Excel. En la Figura 4.2 se muestra un fichero Excel con los valores de los *momentos espectrales* para las secuencias analizadas.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Secuencia	Espectral 0	Espectral 1	Espectral 2	Espectral 3	Espectral 4	Espectral 5	Espectral 6	Espectral 7	Espectral 8	Espectral 9	Espectral 10
2	1TNF	195	245.07	369.51	655.75	1358.52	3240.55	8681.62	25345.66	78466.46	252396.44	832052.58
3	AAA29796	166	173.69	224.51	342.67	599.11	1175.47	2545.55	5987.55	15054.6	39862.59	109747.49
4	AAA29797	177	171.99	213.87	322.05	563.86	1116.27	2445.37	5815.75	14755.99	39337.33	108813.95
5	CAA77743	175	172.03	214.58	322.99	564.89	1117.32	2446.43	5816.82	14757.08	39338.44	108815.09
6												
7												
8												
9												
10												
11												
12												
13												

Figura 4.2. Fichero Excel, resultado del cálculo de los *momentos espectrales*

4.3 Visualización de las formas de representación gráfica de genes y proteínas

ESPECTRO permite obtener representaciones gráficas de genes y proteínas en mapas de colores y en sistemas cartesianos de dos dimensiones. En el menú *Gráfico* de la aplicación principal se acceden a dichas funcionalidades.

4.3.1 Interfaz visual para representaciones en mapas de colores

En la Figura 4.3 se muestra la interfaz para representaciones gráficas de genes y proteínas en mapas de colores. El visual lo compone el área de visualizaciones en la parte izquierda, un panel con las asignaciones de los colores a cada uno de los monómeros que conforman el ADN y las proteínas a modo de leyenda y un panel en la parte superior derecha donde se muestra información del gráfico producto de la interacción del usuario con el mismo.

En la Figura 4.3 se ha representado la proteína *Penicillium chrysogenum* en un mapa de colores. Su cadena de aminoácidos se compone de la siguiente manera:

LSAEQKQQLEEWNNTDGEYPSSKRLHHLIEEVVERHEDKIAVVCDERELTYGELNAQ
GNSLARYLRSIGILPEQLVALFLDKSEKLIVTILGVWKSGAAYVPIDPTYPDERVRFVLD
DTKARAIIASNQHVERLQREVIGDRNLCIIRLE

Se ha visualizado la primera etapa, donde se pliega la secuencia de la proteína dentro de una cuadrícula en forma de espiral. Como se representa la proteína en un mapa de cuatro colores, el conjunto de los veinte aminoácidos se ha dividido en cuatro grupos: polares, apolares, básicos y ácidos. Los polares se han representado de color azul, los apolares de color rojo, los básicos de color verde y los ácidos de color amarillo.

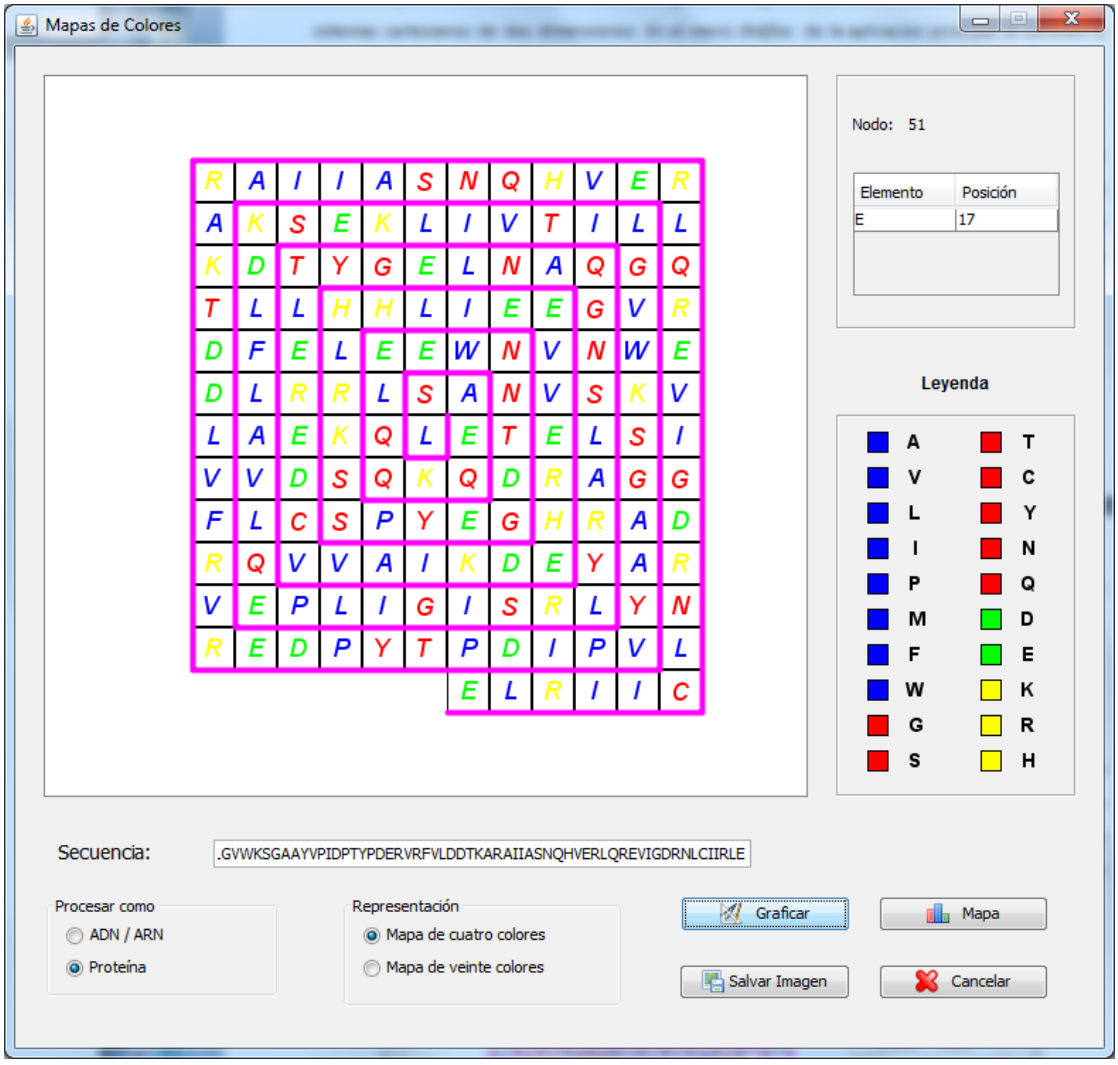


Figura 4.3. Representación en mapa de cuatro colores de la proteína *Penicillium chrysogenum* en su primera etapa

Cuando el usuario presiona el botón *Mapa,* el gráfico que se obtiene se muestra en la Figura 4.4. Se puede observar cómo se han agrupado celdas adyacentes de igual color para formar las distintas regiones que integran el mapa. Cuando el usuario presiona clic sobre una región en el panel superior de la esquina derecha se indica el monómero que se encuentra en esa cuadrícula y su posición dentro de la secuencia. También en la etiqueta *Nodo* se refleja la numeración que tiene asignada la región cuando se conforma el grafo que se deriva de dicha representación gráfica.

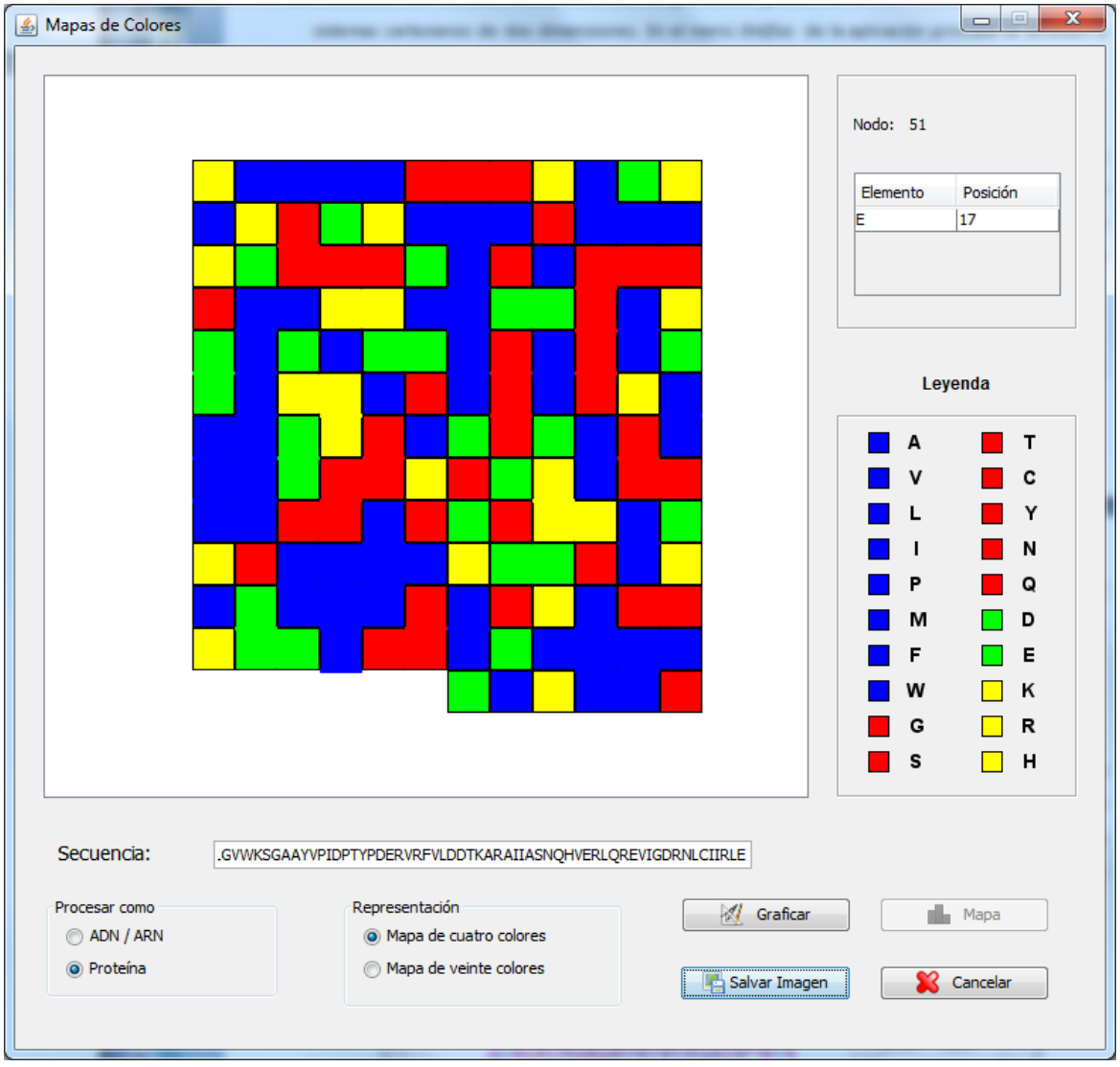


Figura 4.4. Representación en mapa de cuatro colores de la proteína *Penicillium chrysogenum* en su segunda etapa

4.3.2 Interfaz visual para representaciones cartesianas

En la Figura 4.5 se muestra la interfaz para representaciones gráficas de genes y proteínas en sistemas cartesianos de dos dimensiones. En el panel de visualizaciones se ha ploteado la secuencia:

GERLDPALVRRVRESAPHQPAVAHLSSATPSGPWTTCLETGDLAPEWRSVPVGA

que es un fragmento de la proteína *Streptomyces verticillus*.

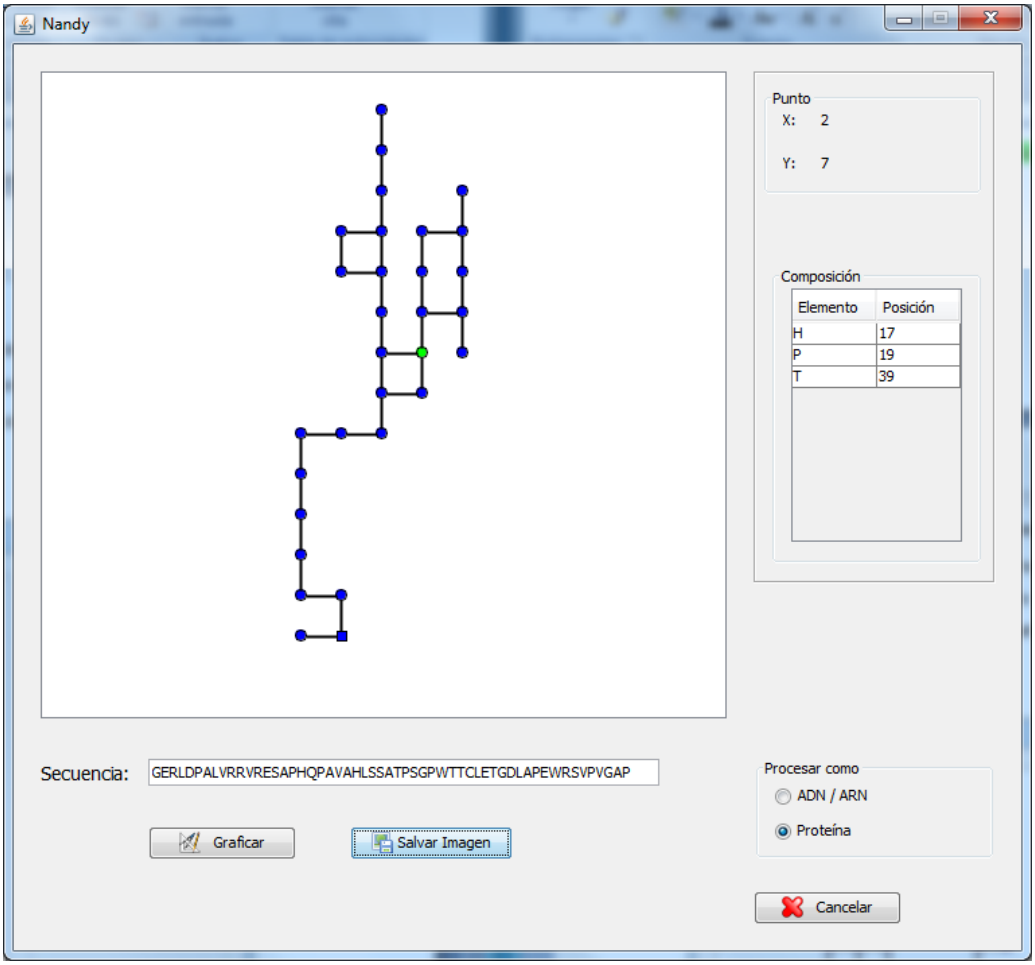


Figura 4.5. Representación cartesiana de un fragmento de la secuencia de aminoácidos de la proteína *Streptomyces verticillus*

Al usuario hacer clic sobre un punto del trazado, este cambia a color verde y se muestra en el panel de la derecha de la ventana la información asociada al nodo del grafo que pertenece dicho punto. La información mostrada son las coordenadas del punto y el conjunto de nucleótidos o aminoácidos que lo componen en dependencia del tipo de secuencia analizada. También se indica la posición que ocupa cada monómero que integra el nodo dentro de la secuencia. El punto de coordenadas (0,0)

perteneciente al primer elemento de la secuencia se diferencia de los demás puntos que componen el gráfico debido a su forma cuadrada.

4.4 Experimentaciones

La caracterización de secuencias de genes o proteínas mediante los *momentos espectrales* permite construir modelos de clasificación que han sido empleados en la predicción o identificación de clases funcionales a las que pueden pertenecer estas biomoléculas. Para la construcción de dichos modelos se utilizó la herramienta de Aprendizaje Automatizado Weka[1]. En la Tabla 4.1 se muestran algunos de los principales algoritmos de clasificación implementados en Weka.

	Name	Function
Bayes	BayesNet	Learns Bayesian nets
	NaiveBayes	Standard probabilistic Naïve Bayes classifier
	NaiveBayesMultinomial	Multinomial version of Naïve Bayes
	NaiveBayesUpdateable	Incremental Naïve Bayes classifier that learns one instance at a time
Trees	Id3	Basic divide-and-conquer decision tree algorithm
	J48	C4.5 decision tree learner
	LMT	Builds logistic model trees
Rules	DecisionTable	Builds a simple decision table majority classifier
	M5Rules	Obtains rules from model trees built using M5
Functions	GaussianProcesses	Gaussian processes for regression
	LinearRegression	Standard multiple linear regression
	MultilayerPerceptron	Backpropagation neural network
	VotedPerceptron	Voted Perceptron algorithm

[1] http://www.cs.waikato.ac.nz/ml/weka/

Lazy	IBk	k-nearest-neighbors classifier
	KStar	Nearest neighbor with generalized distance function
	LWL	General algorithm for locally weighted learning

Tabla 4.1. Algoritmos de clasificación de Weka.

Las técnicas de clasificación basadas en aprendizaje supervisado permiten obtener un modelo matemático o computacional en función de las características o rasgos que describen a un conjunto de datos de los que se conoce su pertenencia a determinado grupo o clase, con el objetivo de establecer reglas según las cuales se pueda clasificar a una nueva instancia en una de las clases existentes, en dependencia de los valores que tengan sus atributos. Para el caso de estudio que nos ocupa se tiene un conjunto de datos formado por secuencias de ADN o proteínas, donde cada secuencia estará descrita por los valores de sus *momentos espectrales* desde el orden cero hasta el orden quince, y el rasgo objetivo es la clase funcional.

4.4.1 ESPECTRO, aplicación a la predicción de funciones tipo bacteriocina

Las bacteriocinas son toxinas proteináceas producidas y exportadas tanto por bacterias Gram positivas como por Gram negativas para inhibir el crecimiento de especies similares u otras más distantes. Desde su descubrimiento por Granita en 1925, muchas de ellas han sido usadas sucesivamente para inhibir tanto patógenos animales como humanos. Por consiguiente las bacteriocinas han sido utilizadas como preservantes alimentarios y son de interés para el desarrollo de nuevos antibióticos; además se ha sugerido su utilización como agentes diagnósticos para el tratamiento del cáncer.

La familia de las bacteriocinas incluye una diversidad de proteínas en términos de tamaño, mecanismo de acción, de inmunidad, liberación, forma de producción, genética y diana microbiana. Debido a su alta diversidad la clasificación de las bacteriocinas ha sido un desafío. La forma clásica de identificación de las bacteriocinas incluye la determinación de la inhibición del crecimiento de otra bacteria por la cepa productora. Pocos métodos bioinformáticos se han desarrollado para identificar supuestos genes de bacteriocinas y proteínas (Dirix, 2004, Kemperman, 2003, Garneau et al., 2002).

Hasta ahora los diferentes métodos bioinformáticos reportados para identificar posibles bacteriocinas se basan en procedimientos de alineamiento de secuencias; los cuales actúan erráticamente en su

clasificación debido a la divergencia existente entre los miembros de esta familia. La aplicación de dichos procedimientos para la detección de bacteriocinas requiere de la implementación de estrategias complejas, además de que presentan limitaciones para usuarios normales al examinar grandes bases de datos de secuencias. Como alternativa a las metodologías existentes se evalúo el comportamiento de la caracterización por los *momentos espectrales* de las secuencias de proteínas calculados por la herramienta ESPECTRO, al utilizar estos conjuntos de datos en el Weka y aplicar técnicas para clasificarlas en bacteriocinas proteináceas o no.

Debido a que las propiedades de hidrofobicidad y basicidad son vitales para la actividad de las bacteriocinas (Hammami et al., 2007)se calcularon los *momentos espectrales* asociados a secuencias plegadas artificialmente en un enrejado cartesiano de Hidrofobicidad y Polaridad (2D-HP). En la Figura 4.6 aparece representada la *colicina E1*, una bacteriocina producida por la E. coli.

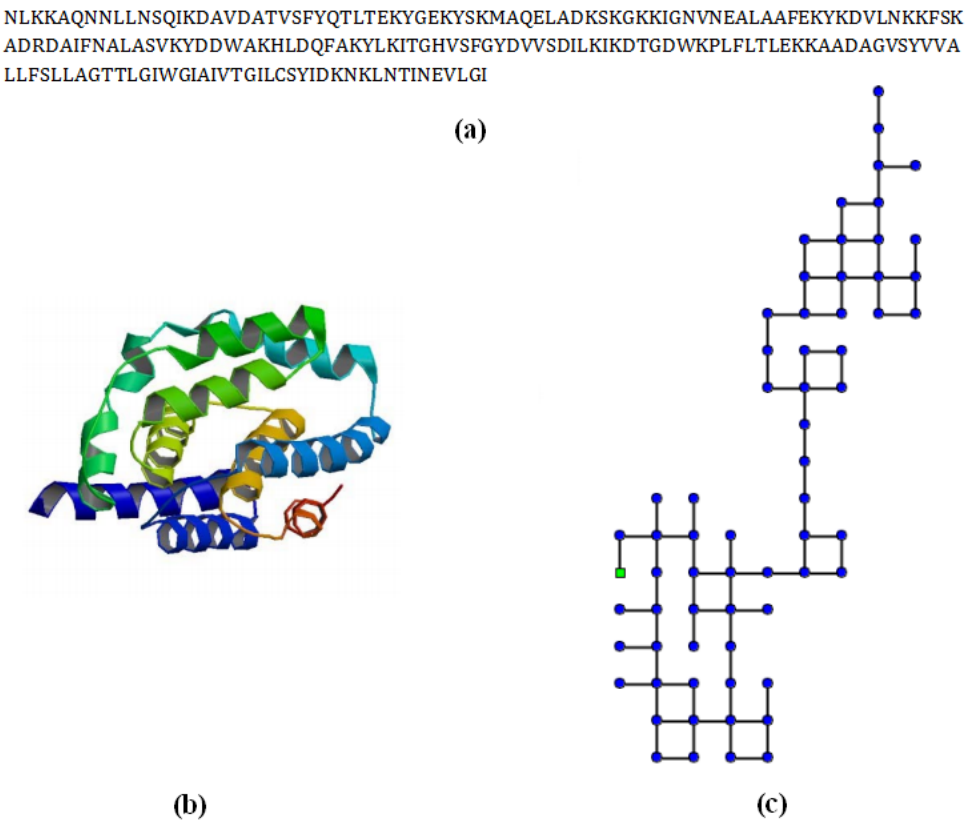


Figura 4.6. Tres estructuras para la secuencia de la colicina E1. (a) Estructura primaria (b) Estructura tridimensional (c) Estructura cartesiana pseudo-secundaria de hidrofobicidad y polaridad.

Los 191 aminoácidos de la secuencia de la *colicina E1* son reorganizados en una estructura pseudo-secundaria obtenida de la representación de la misma en un mapa cartesiano tipo Nandy.

Las direcciones del plano cartesiano estuvieron asociadas a las propiedades de hidrofobicidad y polaridad de los aminoácidos; para ello el conjunto de los 20 aminoácidos que forman las proteínas fueron reagrupados en cuatro clases determinadas por la clasificación que se hace de los mismos en polares, no polares, básicos y ácidos, según se muestra en la Tabla 1.3. La matriz de adyacencia de enlace asociada al grafo que describe la estructura pseudo-secundaria de las proteínas fue ponderada con las cargas electrostáticas de los aminoácidos para el campo de fuerza Amber 95.

Para evaluar la calidad de los *momentos espectrales* calculados por ESPECTRO y su repercusión en problemas de predicción, se colectaron un total de 196 secuencias de proteínas tipo bacteriocinas obtenidas principalmente de las bases de datos BAGEL y BACTIBASE , y 771 secuencias no bacteriocinas descargadas de la base de datos de familias estructurales de proteínas CATH (http://www.cathdb.info). Se consideró polipéptido o proteína teniendo en cuenta la longitud de la secuencia (≥ 100 pb).

Se calcularon los 16 *momentos espectrales* a las 967 secuencias, constituyendo estos los 16 rasgos predictores que caracterizaron estas instancias. Se construyó un fichero de entrada para el Weka (*.arff) con los datos obtenidos y se procedió a su análisis mediante los distintos métodos de clasificación que posee la herramienta.

A continuación se describen los resultados para los modelos que mostraron mejor desempeño. En todos los casos se realizó validación cruzada realizando 10 particiones.

Algoritmo BayesNet

Bact	*No Bact*	*Razón*	*%*
125	71	125/196	79.00
132	**639**	639/771	

El modelo mediante redes bayesianas clasificó correctamente a 125 de 196 bacteriocinas y a 639 de 771 proteínas no bacteriocinas. Globalmente el modelo clasificó correctamente a 754 secuencias de un total de 967, para un desempeño de un 79 %.

Algoritmo Lazy.LWL

Bact	*No Bact*	*Razón*	*%*
146	50	146/196	84.17
103	**668**	668/771	

El modelo obtenido con el algoritmo de aprendizaje de ponderación local (Locally Weighted Learning; LWL) clasificó correctamente a 146 de 196 bacteriocinas y a 668 de 771 proteínas no bacteriocinas. Globalmente el modelo clasificó correctamente a 814 secuencias de un total de 967, para un desempeño de un 84.17 %.

Algoritmo IBK

Bact	*No Bact*	*Razón*	*%*
107	89	107/196	87.07
36	**735**	735/771	

El modelo obtenido empleando el algoritmo de los k-vecinos más cercanos fue el que mostró mejor comportamiento global, clasificó correctamente a 845 secuencias de un total de 967 para un desempeño de un 87.07 %. Específicamente clasificó correctamente a 107 de 196 bacteriocinas y a 735 de 771 proteínas del grupo de control. Notar sin embargo, que la precisión en el grupo positivo fue algo más baja que en los dos modelos anteriores. Se realizaron los experimentos para un *K*=12.

Además, se compararon estos resultados con otras fuentes clásicas de anotación predictiva funcional, las 196 bacteriocinas proteicas del estudio fueron analizadas usando la herramienta InterProScan. Este último combina diferentes métodos de reconocimiento de clases de proteínas propios para los miembros de la base de datos InterPro en un solo recurso, con búsqueda de la correspondiente anotación de InterPro y GO (Gene Ontologhy). Muchos de los métodos de reconocimientos de clases de proteínas implementados en InterPro están basados en cierto grado, en procedimientos de alineamiento, lo cual justifica el por qué se ha seleccionado para llevar a cabo un estudio comparativo con nuestro método libre de alineamiento.

En este sentido, la herramienta InterProScan no clasificó 40 secuencias de proteínas de un total de 196. De estas 40 secuencias no clasificadas, 16 no fueron reconocidas (no hits) y el resto no se

asociaron a las clases integradas en la base de datos de InterPro, siendo solamente clasificadas el 79.6% de los datos. Además 38 bacteriocinas proteináceas fueron reconocidas por InterPro como otras clases proteicas no relacionadas a secuencias tipo bacteriocinas, disminuyendo el porcentaje de buena clasificación a 60.2 % (118/196).

Se puede concluir entonces, que a pesar de no ser mucha la diferencia significativa, los modelos obtenidos mediante el cálculo de los *momentos espectrales* tuvieron un mejor desempeño (146/196) que los arrojados por la herramienta InterProScan (118/196).

4.4.2 Identificación de dominios de adenilación (A) mediante mapas de colores.

El desempeño del software también se evalúo en la identificación de dominios de adenilación (A) pertenecientes a las péptido-sintetasas no-ribosomales (Nonribosomal peptide synthetases: NRPS). Para el estudio de los dominios de adenilación (A), las secuencias de proteínas pertenecientes a estos dominios fueron representadas gráficamente utilizando mapas de 20 colores y caracterizadas numéricamente a través de los *momentos espectrales* derivados de dichas representaciones.

Las péptidos-sintetasas no-ribosomales (NRPS) son proteínas que están organizadas en unidades funcionales interactivas denominadas módulos, los cuales catalizan un conjunto de reacciones catalíticas que llevan a la formación de un péptido (Doekel y Marahiel, 2001). Los biocatalizadores NRPS son responsables de la síntesis de un enorme número de metabolitos secundarios con diversa estructura y funciones. Muchos de estos compuestos activos sintetizados por los NPRS resultan de gran interés para la química médica debido a su actividad anti-microbiana, anti-parasitaria, anti-tumoral y como agentes inmunosupresores. Se necesitan tres dominios para un módulo NRPS básico: un dominio de adenilación (A) que selecciona el aminoácido y lo activa como adenilato de sodio, un dominio peptidil carrier de proteína (peptidyl carrier protein: PCP o dominio T) que fija el cofactor 4' fosfopanteteína y al cual se une el aminoácido y un dominio de condensación (C) que cataliza la formación de la unión peptídica.

El la Figura 4.7 se muestra el complejo proteico NPRS que sintetiza el *bacitracin*, un antibiótico que se indica en contra de bacterias Gram positivas, especialmente en heridas y mucosas, porque inhibe la formación de la pared celular de dichos microorganismos. En la parte inferior izquierda de la figura se muestra la distribución de los diferentes dominios por los cuales está compuesto este complejo proteico. También en la Figura 4.7 se muestra la secuencia de aminoácidos que integran el primer dominio de adenilación (A) del complejo multimodular.

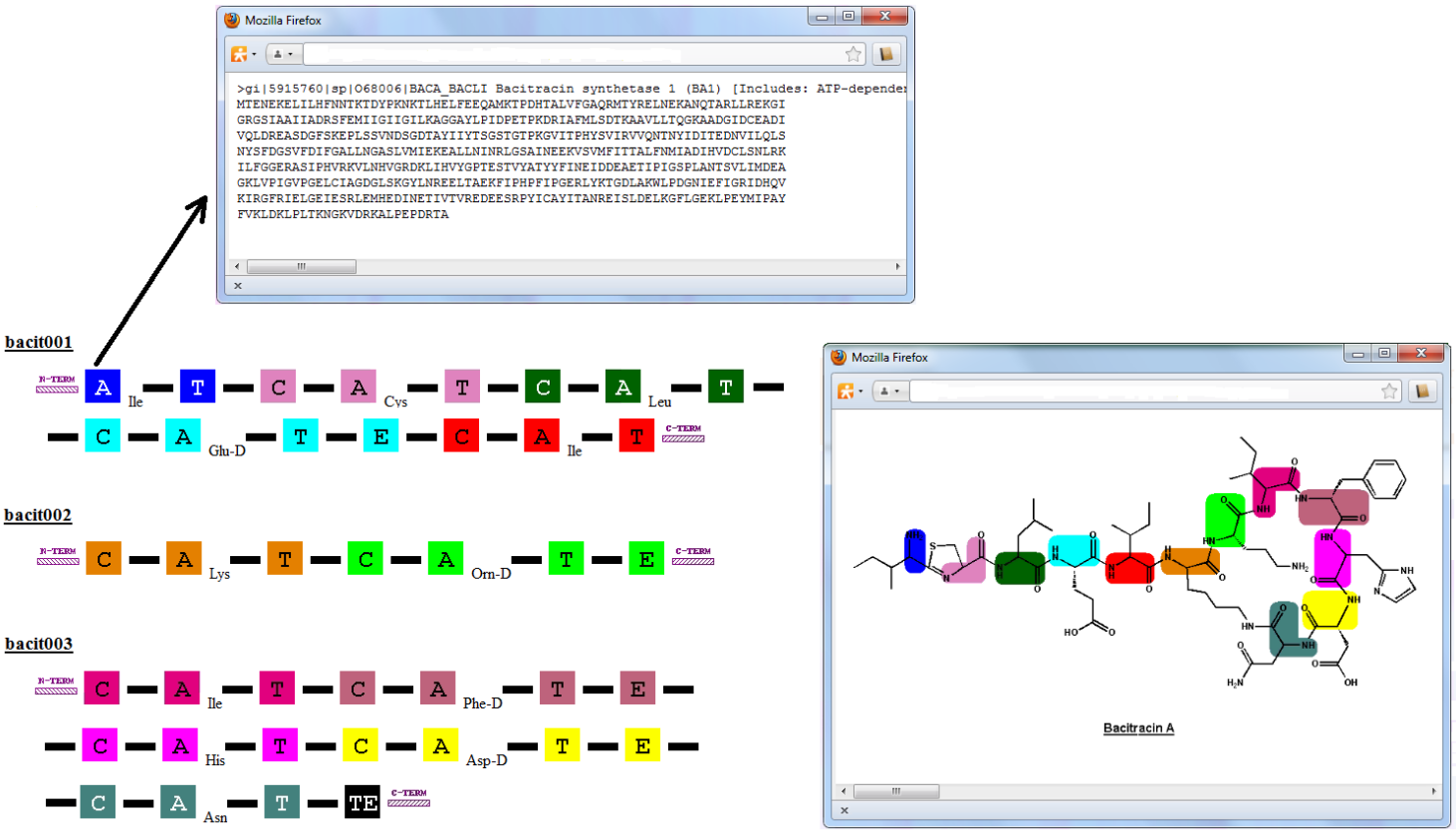


Figura 4.7. Complejo proteico NPRS que sintetiza el bacitracin.

Los dominios (A) muestran una alta divergencia en su secuencia con respecto a los miembros de su familia; por lo que se hace difícil su identificación mediante procedimientos clásicos de alineamiento. De hecho, este dominio no se encuentra en la base de datos de dominios conservados del NCBI (National Center for Biotechnology Information). A pesar de estas evidencias, enfoque libres de alineamiento basados en support vector machine (SVM) se han ocupado de la predicción de los dominios (A), particularmente para tipos de sustrato específicos. Como alternativa a la metodología existente se evalúo la precisión de los modelos de clasificación obtenidos por la herramienta Weka a partir de los *momentos espectrales* calculados por ESPECTRO en la identificación de dominios de adenilación (A) entre la diversidad de clases y dominios de clasificación pertenecientes a la base de datos de familias estructurales de proteínas CATH (Class, Architecture, Topology and Homology).

Para la prueba se colectaron un total de 721 secuencias de proteínas, divididas en 147 pertenecientes a dominios de adenilación, descargadas de la mayor base de datos existentes para los mismos, la

NRPS–PKS, disponible en (http://www.nii. res.in/nrps-pks.html) y 574 secuencias de proteínas no identificadas como dominios (A), descargadas de la base de datos de familias estructurales de proteínas CATH (http://www.cathdb.info). Particularmente se utilizó la base de datos de secuencias FASTA para todos los dominios CATH basados en datos de secuencias COMBS[2].

A las 721 secuencias de proteínas analizadas se le calcularon los *momentos espectrales* desde el orden cero hasta el orden quince. Cada secuencia quedó descrita por 16 rasgos predictores más su atributo de clase. Se construyó un fichero de entrada para el Weka (*.arff) con los datos obtenidos y se procedió a su análisis mediante los distintos métodos de clasificación que posee la herramienta.

A continuación se describen los resultados para los modelos que mostraron mejor desempeño. En todos los casos se realizó validación cruzada realizando 10 particiones.

Algoritmo Lazy.LWL

Dominio (A)	*No Dominio*	*Razón*	*%*
144	3	144/147	99.16
3	**571**	571/574	

El modelo obtenido con el algoritmo de aprendizaje de ponderación local clasificó correctamente a 144 de 147 secuencias de proteínas que identifican dominios (A) y a 571 de 574 secuencias del grupo de contraste. Globalmente el modelo clasificó correctamente a 715 secuencias de un total de 721, para un desempeño de un 99.16 %.

[2] Cuando se trata de la secuencia de aminoácidos de un dominio dado CATH (o cadena de aminoácidos de un fichero PDB), es importante diferenciar entre la secuencia de aminoácidos tal como se define por la secuencia de registros ATOM y el definido por los registros SEQRES en el fichero PDB (no son necesariamente los mismos). Para resolver ese problema la base de datos de familias estructurales de proteínas CATH emplea un algoritmo interno que alinea las secuencia SEQRES y ATOM y provee una secuencia completa de aminoácidos a la que denomina secuencia COMBS.

Algoritmo MultilayerPerceptron

Dominio (A)	*No Dominio*	*Razón*	*%*
145	2	145/147	99.16
4	**570**	570/574	

El modelo obtenido mediante redes neuronales tipo MultilayerPercentron clasificó correctamente a 145 de 147 secuencias de proteínas que identifican dominios (A) y a 570 de 574 secuencias del grupo de contraste. Globalmente el modelo clasificó correctamente a 715 secuencias de un total de 721, para un desempeño de un 99.16 %. La topología de la red que se especificó fue de una capa oculta con nueve neuronas.

Algoritmo IBK

Dominio (A)	*No Dominio*	*Razón*	*%*
146	1	146/147	99.58
2	**572**	572/574	

El modelo obtenido empleando el algoritmo de los k-vecinos más cercanos clasificó correctamente a 146 de 147 secuencias de proteínas que identifican dominios (A) y a 572 de 574 secuencias del grupo de contraste. Globalmente fue el que mejor desempeño mostró, clasificó correctamente a 718 secuencias de un total de 721, para un 99.58 %. Se realizaron los experimentos para un *K*=7.

Como resultados de estas pruebas en la identificación de dominios (A) se puede concluir que los valores de los *momentos espectrales* calculados por la herramienta ESPECTRO aportan información útil que cuantifica la esencia de la composición y distribución de las secuencias proteicas que constituyen dominios de adenilación.

4.5 Conclusiones parciales

El software ESPECTRO tiene una interfaz amigable y permite a los bioinformáticos realizar experimentaciones con biomoléculas descritas en varios formatos y considerando diversas formas de representaciones gráficas.

Las representaciones y consecuentes caracterizaciones logradas con ESPECTRO aportan información útil que cuantifica la esencia de la composición y distribución de las secuencias proteicas que constituyen dominios de adenilación y en la predicción de funciones tipo bacteriocina.

CONCLUSIONES Y RECOMENDACIONES

Se desarrolló el software ESPECTRO para el análisis por métodos gráficos de secuencias de genes y proteínas que permita caracterizarlas numéricamente a través de índices topológicos, dando cumplimiento así al objetivo general propuesto, ya que:

- Se identificaron los principales métodos gráficos empleados en la representación de secuencias de ADN y proteínas: entre ellas las cartesianas, las que representan al ADN como espectro, la representación de proteínas en grafos tipo estrella, las proteínas como matrices de adyacencia de aminoácidos y los mapas de cuatro colores. Cada una de estas representaciones permite extraer información de las secuencias desde distintos enfoques y posibilitan que fragmentos con información relevante se puedan obtener rápidamente por la inspección visual de la trama de la secuencia, permiten establecer una relación entre la estructura de la molécula y su actividad y son libres de alineamientos.
- El software ESPECTRO desarrollado permite caracterizar numéricamente secuencias de ADN y proteínas a partir de las representaciones gráficas: cartesiana de Nandy y los mapas de colores, aunque su diseño es extensible y permite incorporar otras formas de representación.
- En ESPECTRO, un especialista en Bioinformática, puede realizar un ciclo cerrado de análisis de las biomoléculas transitando desde cargar los datos en diferentes formatos, graficar las biomoléculas incluyendo varios tipos de representaciones, calcular los *momentos espectrales* a partir de la representación seleccionada y salvar los resultados para ser utilizados en otros sistemas que permitan predecir otros comportamientos.
- Las representaciones y consecuentes caracterizaciones logradas con ESPECTRO aportan información útil que cuantifica la esencia de la composición y distribución de las secuencias proteicas que constituyen dominios de adenilación y en la predicción de funciones tipo bacteriocina.

Referencias bibliográficas

AGÜERO-CHAPIN, G., MOLINA-RUIZ, R., SÁNCHEZ-RODRÍGUEZ, A. & ANTUNES, A. 2011a. Non-linear models based on simple topological indices to identify RNase III protein members. *Journal of Theoretical Biology,* 273**,** 167 - 178.

AGÜERO-CHAPIN, G., PÉREZ-MACHADO, G., MOLINA-RUIZ, R. & ANTUNES, A. 2010. TI2BioP: Topological Indices to BioPolymers. Its practical use to unravel cryptic bacteriocin-like domains. *Amino Acids,* 40**,** 431 - 442.

AGÜERO-CHAPIN, G., SÁNCHEZ-RODRÍGUEZ, A., MOLINA-RUIZ, R. & ANTUNES, A. 2011b. An Alignment-Free Approach for Eukaryotic ITS2. Annotation and Phylogenetic Inference. *PLoS One: e26638,* 6.

BALASUBRAMANIAN, K. 1985. Applications of combinatorics and graph theory to spectroscopy and quantum chemistry. *Chem. Rev,* 6**,** 599 - 618.

BONDY, J. A. & MURTY, U. S. R. 1976. *Graph Theory with Applications,* North-Holland.

BROWN, T. 1999. *Genetics: A molecular approach,* Garland Science.

CARDELLÁ & HERNÁNDEZ 1999. *Bioquímica Médica*.

DIRIX, G. 2004. Peptide signal molecules and bacteriocins in Gram-negative bacteria: a genome-wide in silico screening for peptides containing a double-glycine leader sequence and their cognate transporters. *Peptides,* 25**,** 1425.

ESTRADA, E. 1996. Spectral moments of the edge adjacency matrix in molecular graphs. 1. Definition and applications to the prediction of physical properties of alkanes. *Chem Inf Comput Sci,* 36**,** 844-849.

ESTRADA, E. 1997. Spectral moments of the edge-adjacency matrix of molecular graphs. 2. Molecules containing heteroatoms and QSAR applications. *Chem Inf Comput Sci,* 37**,** 320-328.

ESTRADA, E. 1998. Generalized Spectral Moments of the Iterated Line Graphs Sequence. A Novel Approach to QSPR Studies. *J. Chem. Inf. Comput. Sci.,* 39**,** 90 - 95.

FOWLER, M. & SCOTT, K. (eds.) 1997. *UML Distilled,* Massachusetts: Addison Wesley Longman.

GARNEAU, S., MARTIN, N. & VEDERAS, J. 2002. Two-peptide bacteriocins produced by lactic acid bacteria. *Biochimie,* 84**,** 577 - 592.

GATES, M. A. 1985. Simple DNA sequence representations. *Nature,* 316**,** 219.

GONZALEZ-DIAZ, H. 2007. ANN-QSAR model for selection of anticancer leads from structurally heterogeneous series of compounds.

HAMMAMI, R., ZOUHIR, A., HAMIDA, J. B. & FLISS, I. 2007. BACTIBASE: a new web-accessible database for bacteriocin characterization. *BMC Microbiology,* 7**,** 89.

HARARY, F. 1969. *Graph Theory,* Addison-Wesley.

HEILEMAN, G. L. 1998. *Estructuras de datos, algoritmos, y programación orientada a objetos*, Mc Graw-Hill.

JACOBSON, I., BOOCH, G. & RUMBAUGH, J. (eds.) 2000. *El Proceso Unificado de Desarrollo de Software*: Addison Wesley.

JUNGNICKEL, D. 2008. *Graphs, Networks and Algorithms*, Springer.

KEMPERMAN, R. 2003. Identification and characterization of two novel clostridial bacteriocins, circularin A and closticin 574. *Appl Environ Microbiol*, 69**,** 1589.

KOURÍ, J. B. 1978. *Biología General 1*.

LEHNINGER, A. L. 1987. *Principios de Bioquímica*, Prentice Hall.

LEONG, P. M. & MOGENTHALER, S. 1995. Random walk and gap plots of DNA sequences. *Comput. Appl. Biosci*, 12**,** 503-511.

MATHEWS, HOLDEN, V. & AHERN 1999. *Biochemistry*, Prentice Hall.

MCCALL, J. J. 1985. An Introduction to Exchangeability and Its Economic Applications.

MUGHAL, K. A. & RASMUSSEN, R. W. (eds.) 2004. *Programmer's Guide to Java*, Boston: Addison Wesley.

NANDY, A. 1994. A new graphical representation and analysis of DNA sequence structure. I. Methodology and application to globin genes. *Current Science,* 66**,** 309 - 314.

NANDY, A., HARLE, M. & BASAK, S. C. 2006. Mathematical descriptors of DNA sequences: development and applications

RANDIC, M. & BALABAN, A. T. 2005. Four-color map representation of DNA or RNA sequences and their numerical characterization. *Chemical Physics Letters,* 407**,** 205-208.

RANDIC, M., MARJAN, V., NOVIC, M. & B.SUBHASH 2000. On ordering of folded structures *Communications in Mathematical and in Computer Chemistry,* 42**,** 181 - 231.

RANDIC, M., VRACKO, M., LERS, N. & PLAVSIC, D. 2003. Analysis of similarity/dissimilarity of DNA sequences based on novel 2-D graphical representation. *Chemical Physics Letters,* 371**,** 202 - 207.

RANDIC, M., ZUPAN, J., BALABAN, A. T. & PLAVSIC, D. 2010. Graphical Representation of Proteins. *American Chemical Society,* 111**,** 790-862.

RANDIC, M., ZUPAN, J. & VIKIC-TOPIC 2006. On representation of proteins by star-like graphs. *Journal of Molecular Graphics and Modelling,* 26**,** 290 - 305.

RAYCHAUDHURY, C. & NANDY, A. 1999. Indexing scheme and similarity measures for macromolecular sequences. *J. Chem. Inf. Comput. Sci.,* 39**,** 243 - 247.

RUMBAUGH, J. & BOOCH, G. (eds.) 2000. *El Lenguaje Unificado de Modelado. Manual de Referencia*: Addison Wesley.

SEMENOVICH, B. 2004. *The Physics Of Traffic:Empirical Freeway Pattern Features, Engineering Applications, And Theory*, Springer.

VENKATA, R. 2007. *Decision Making in the Manufacturing Environment:Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*, Springer.

ZHANG, Y., WANG, Z. & ZHANG, J. 2010. Research Progress of Complex Electric Power Systems: Graph Theory Approach. *International Journal of Electrical Power & Energy Systems,* 14**,** 254 - 260.