UNIVERSIDAD DE SEVILLA


Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

# *TrLab*: Una metodología para la extracción y evaluación de patrones de comportamiento de grandes volúmenes de datos biológicos dependientes del tiempo

TESIS DOCTORAL

Autor:

David Gutiérrez Avilés

Directora:

Cristina Rubio Escudero

Sevilla, Junio de 2015

*A Ana.*
*A Papá.*
*A Mamá.*
*A Loren.*
*A Magui.*

# Agradecimientos

En primer lugar quisiera dar las gracias a la principal persona por la que, sin ella, no hubiera sido posible la realización de esta investigación, mi directora, Cristina Rubio Escudero. Cristina no sólo me ha dirigido, animado, asesorado y ayudado en la elaboración de este trabajo, aspectos que se le presuponen a cualquier director de tesis que se precie; Cristina me ha enseñado a analizar todos los aspectos de mi trabajo desde un punto de vista crítico, a redactar artículos científicos, a aplicar exhaustivamente las metodologías científicas, a no darme por vencido ante un revés, a interpretar correctamente todos los vaivenes de la investigación. Desde mi humilde punto de vista, es la mejor directora que un alumno de doctorado puede tener. Gracias Cristina.

Al Departamento de Ingeniería Eléctrica de la Universidad de Sevilla y en especial a José Antonio Rosendo por darme la oportunidad de formar parte de su equipo, de aprender y enriquecerme de metodologías y entornos de trabajo distintos a los de mi campo de investigación y de proporcionarme un puesto de trabajo sin el cual me habría sido muy difícil la consecución de esta tesis. Pero, sobre todo, por haberme dado la oportunidad de conocer, convivir y compartir vivencias con mis amigos Manolo Barragán, Manolo Nieves, Javier Serrano, Cristina Carmona, Juan Jiménez y Francisco de Paula; ellos, aún siendo colegas investigadores de otros campos, han sido mis auténticos compañeros a lo largo de todo este viaje.

Al Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla por darme la oportunidad de formar parte del programa de doctorado en Ingeniería Informática y por darme el soporte para poder publicar en revistas y congresos científicos. Agradecer también a todos los compañeros de mi grupo de investigación del departamento por su disponibilidad para proporcionarme ayuda siempre que la he necesitado. Quiero hacer una mención especial a Francisco Martínez de la Universidad Pablo de Olavide por apoyarme de manera cercana en mis primeros pasos en la difusión de los resultados de investigación.

Quiero dar las gracias al otro pilar fundamental e indispensable para la realización de esta investigación: a mi familia. Gracias a Ana por estar en todo momento a mi lado, por tu cariño y comprensión, por tus ánimos, por no dejarme desfallecer y por ser mi guía. Gracias Papá y Mamá por haberme permitido realizar esta tesis y por haberme apoyado en todo momento a todos los niveles y por haberme inculcado todos los valores indispensables para llevar a cabo este trabajo. Gracias a mi hermano Loren por su incansable apoyo y por contagiarme siempre su buen talante, su buen humor y su inmensa bondad. Y gracias al ser que me ha hecho compañía inquebrantable, fiel y constantemente cada minuto del solitario trabajo de escritura de tesis doctoral, mi perra Magui.

Por último, quiero agradecer también a otro de lo componentes fundamentales en mi plano personal: mis amigos de toda la vida. Gracias a Juanki, Inma, Julio, Inma, Igna, Rocío, Fiti, Bea, Juan, Tati y Álvaro por vuestro apoyo, dedicación y comprensión.

MUCHAS GRACIAS A TODOS!

# Resumen

La tecnología de microarray ha revolucionado la investigación biotecnológica gracias a la posibilidad de monitorizar los niveles de concentración de ARN. El análisis de dichos datos representa un reto computacional debido a sus características. Las técnicas de Clustering han sido ampliamente aplicadas para crear grupos de genes que exhiben comportamientos similares. El Biclustering emerge como una valiosa herramienta para el análisis de microarrays ya que relaja la restricción de agrupamiento permitiendo que los genes sean evaluados sólo bajo un subconjunto de condiciones experimentales. Sin embargo, ante la consideración de una tercera dimensión, el tiempo, el Triclustering se presenta como la herramienta apropiada para el análisis de experimentos longitudinales en los que los genes son evaluados bajo un cierto subconjunto de condiciones en un subconjunto de puntos temporales. Estos triclusters proporcionan información oculta en forma de patrón de comportamiento para experimentos temporales con microarrays.

En esta investigación se presenta *TrLab*, una metodología para la extracción de patrones de comportamiento de grandes volúmenes de datos biológicos dependientes del tiempo. Esta metodología incluye el algoritmo *TriGen*, un algoritmo genético para la búsqueda de triclusters, teniendo en cuenta de forma simultánea, los genes, condiciones experimentales y puntos temporales que lo componen, además de tres medidas de evaluación que conforman el núcleo de dicho algoritmo así como una medida de calidad para los triclusters encontrados.

Todas estas aportaciones estarán integradas en una aplicación con interfaz gráfica que permita su fácil utilización por parte de expertos en el campo de la biología.

Las tres medidas de evaluación desarrolladas son: $MSR_{3D}$ basada en la adaptación a las tres dimensiones del Residuo Cuadrático Medio, $LSL$ basada en el cálculo de la recta de mínimos cuadrados que mejor ajusta la representación gráfica del tricluster y $MSL$ basada en el cálculo de los ángulos que forman el patrón de comportamiento del tricluster. La medida de calidad se denomina $TRIQ$ y aglutina todos los aspectos que determinan el valor de un tricluster: calidad de correlación, gráfica y biológica.

**Palabras clave** triclustering, algoritmos genéticos, microarrays, datos temporales, minería de datos, bioinformática

# Índice general

# Índice de figuras

# Índice de tablas

# Parte I

# MEMORIA

# Capítulo 1

# Introducción

Este capítulo nos proporcionará el fondo necesario para el correcto entendimiento del contenido de esta tesis doctoral. En primera instancia se realizará una descripción del problema que se plantea en esta investigación junto con un análisis de las propuestas realizadas en el campo (Sección 1.1). Seguidamente justificaremos las razones que nos han llevado a realizar la presente investigación (Sección 1.2) y los objetivos que se han planteado en la misma (Sección 1.3) finalizando este capítulo de prólogo con una guía que especifique la estructura en secciones del documento (Sección 1.4) así como una breve descripción de su contenido.

## 1.1. Descripción del problema

La tecnología de microarray es ampliamente utilizada en entornos de investigación de orden biológico gracias a su capacidad de monitorizar el nivel de concentración de ARN de un gran grupo de genes; este hecho nos habilita para el estudio de las funciones genéticas de las especies monitorizadas [9]. Con el objetivo de encontrar conocimiento nuevo, válido y oculto a la percepción humana [2, 59] los campos de la Minería de Datos (Data

Mining) y la Bioinformática (Bioinformatics) surgieron desarrollando numerosas herramientas, metodologías y técnicas computacionales que nos permiten analizar grandes volúmenes de datos tanto de fuentes de carácter general como biológicas. Uno de los enfoques más estudiados en las citadas disciplinas es el descubrimiento de patrones de comportamiento en datos de expresión genética.

Se ha comprobado en numerosos trabajos como [72, 39, 73] que los genes que exhiben una alta correlación a través de sus niveles de expresión pueden estar involucrados en procesos reguladores similares, además, la relación entre correlación y funcionalidad ha sido probada en diversos estudios como en [19].

Las técnicas de Clustering son adecuadas para la detección de patrones de comportamiento mediante la creación de grupos de genes que manifiestan patrones de expresión similares [65]. Los algoritmos de Clustering tradicionales realizan un análisis completo del espacio dimensional que ofrece el microarray, agrupando genes teniendo en cuenta todas las condiciones experimentales del mismo [34], sin embargo, la actividad genética puede aparecer bajo un conjunto de condiciones experimentales concretas, exhibiendo patrones de comportamiento locales. El hallazgo de estos patrones locales es clave para descubrir secuencias genéticas que podrían ser costosas de obtener mediante otras vías por lo que el paradigma de las técnicas de Clustering deben ser extendidas a métodos que permitan el descubrimiento de patrones de comportamiento locales en datos de expresión genética [6]. Las técnicas de Biclustering [13] atacan este problema relajando el método de agrupamiento, permitiendo la asociación de genes bajo únicamente un subconjunto de condiciones experimentales, demostrando éxito en la búsqueda de patrones de comportamiento genéticos [42, 58].

Avanzando un paso más en cuanto a la dimensionalidad de los experimentos con microarray, existe un gran interés en experimentos temporales con esta tecnología puesto que permiten un análisis en profundidad de procesos moleculares en los que la evolución temporal es importante, por ejemplo, los ciclos celulares, el desarrollo a nivel molecular o la evolución

de enfermedades [3]. Ésto supone el añadido de una tercera dimensión a los microarrays además de las ya citadas dimensiones de genes y condiciones experimentales; ésta es, la dimensión de tiempo, lo que implica que las metodologías de Clustering y Biclustering son insuficientes para atacar esta nueva coyuntura.

En este sentido, surge la técnica del Triclustering dando un paso más y permitiendo agrupar genes bajo unas condiciones experimentales concretas y bajo unos puntos temporales concretos [43], por consiguiente, habilitando la posibilidad de analizar datos 3D. Por lo tanto, el Triclustering es adecuado para el análisis de experimentos de microarray en el que varias muestras son adquiridas en diferentes puntos temporales [20]; ésto supone un hecho de gran interés ya que permite un análisis acentuado de procesos biológicos donde el desarrollo temporal es importante.

Tanto las técnicas de Biclustering como las de Triclustering atacan problemas NP-completos [74]; en consecuencia, los algoritmos basados en metaheurísticas, se antojan indispensables para solucionar este tipo de problema. En este sentido, definir una medida apropiada de calidad de los productos del Triclustering o triclusters es un reto importante y esencial [21].

Tras el análisis profundo de los trabajos recientes en el campo del análisis de datos temporales de expresión genética y en particular aquellas investigaciones relacionadas con el desarrollo y aplicación de técnicas de Triclustering, centramos la atención, entre otros, en [82] mediante el cual los autores Zhao y Zaki introducen el algoritmo *triCluster*. Ésta es una de las primeras metodologías para extraer patrones de comportamiento en datos de expresión genética en 3D basada en la medición de la calidad de los triclusters fundamentado en su simetría. Este enfoque permite una extracción de triclusters muy eficiente ya que son buscados en las dimensiones de menos cardinalidad. Los triclusters tienen que satisfacer algunos requerimientos: maximalidad, es decir, no existe un tricluster en el conjunto de soluciones totalmente incluido en otro tricluster de dicho conjunto; delimitación, dada por $\epsilon$, del ratio para los valores de cada par de columnas en el tricluster y

la determinación del volumen máximo y mínimo de los triclusters dada por la relación entre $\delta^x$, $\delta^y$, $\delta^z$ para los genes, condiciones y puntos de tiempo respectivamente.

Un año después, fue publicada una versión extendida y generalizada de la propuesta de Zhao y Zaki, llamada *g-triCluster* [36]; en ella, los autores aducen que la propiedad de simetría no es válida para todos los patrones presentes en datos biológicos y propusieron el índice de correlación de Spearman [70] como una medida de evaluación de triclusters más apropiada.

Una propuesta basada en computación evolutiva fue presentada en [40], en ella, la función de fitness es definida como una medida multi-objetivo que intenta optimizar tres factores: tamaño, homogeneidad y varianza en la dimensión de los genes de los triclusters.

La metodología *LagMiner* fue presentada en [80] en la cual el objetivo era encontrar triclusters con retardos temporales que permitía encontrar relaciones a nivel de regulación entre genes; está basada en un novedoso modelo de clusters 3D llamado $S^2D^3$ *Cluster* en el que evalúan sus triclusters acorde a criterios de homogeneidad, regulación, mínimo número de genes, subespacio de condiciones experimentales y longitud de los periodos temporales.

Wang *et al.* [77] propusieron un nuevo algoritmo llamado *ts-cluster* basando su definición en el concepto de tricluster coherente en el que también buscaba relaciones a nivel de regulación entre genes. En este sentido, el desplazamiento es también considerado entre los puntos temporales de los triclusters evaluados.

Una nueva estrategia para extraer clusters 3D en datos con valores reales fue introducida en [69] en la cual los autores definieron los *CSCs* (Correlated 3D Subspace Clusters) donde los valores de cada cluster deben tener un alto nivel de ocurrencia y dichas ocurrencias no debían ser obtenidas por casualidad. Su medida evaluaba los triclusters en base a una medida de correlación que tenía en cuenta los requisitos antes mencionados.

Hu *et al.* presentaron un enfoque haciendo hincapié en el concepto de *Low-Variance 3-Cluster* [35], por el cual, se tenía en cuenta la restricción de la baja varianza en la distribución de los valores celulares.

Otro enfoque basado en encontrar reglas de asociación con dependencia temporal fue presentado en [41]. Las reglas obtenidas representaban relaciones de regulación entre genes.

En 2011 [43] y 2012 [44] fueron publicados sendos estudios de Triclustering aplicado a series temporales de expresión genética. Entre las últimas aportaciones encontramos el trabajo de Tchagang *et al.* [75] y de Gnatyshak [24].

## 1.2. Justificación

Los motivos que nos llevan a realizar esta investigación son diversos. En primera instancia y tras estudiar concienzudamente el abanico de posibilidades que ofrece el estado del arte, llegamos a la conclusión de que el Triclustering aplicado al análisis de datos de expresión genética y, más específicamente, a experimentos de microarray con desarrollo temporal (así como para otras técnicas relacionadas como los datos ChIP-chip [79], repositorios RNA-seq [47], etc) constituye un tema en auge dentro del campo, con gran interés dentro de la comunidad científica y con un potencial muy alto como herramienta de apoyo al experto biológico.

Como fruto del estudio del estado del arte encontramos en segunda instancia pocas propuestas basadas en técnicas de inteligencia artificial tales cómo algoritmos genéticos. Igualmente no apreciamos ninguna propuesta que tenga en cuenta el aspecto gráfico de los triclusters resultantes del proceso siendo, no obstante, un punto clave en la valoración positiva de éstos en todas los trabajos analizados. Asimismo apreciamos una necesidad de establecer una medida de evaluación de triclusters que aglutine todos los aspectos usados para tal efecto en la literatura, estos son, evaluación gráfica, de correlación y biológica.

En última instancia, observamos que estas técnicas tienen potencialidad para ser efectivas en múltiples escenarios en los que las tres dimensiones jueguen un papel importante pero, sin embargo, apreciamos una carestía de trabajos aplicados a datos de origen no biológico.

## 1.3. Objetivos

Las metas a alcanzar con este trabajo son las siguientes:

- Estudio del estado del arte relacionado con el enfoque del Triclustering, análisis de datos de expresión genética y estudios de experimentos de microarray temporales.
- Desarrollo de un algoritmo de Triclustering basado en el paradigma de los algoritmos genéticos (algoritmo $TriGen$).
- Proponer y desarrollar una medida de evaluación de triclusters basada en la adaptación de la medida de Cheng y Church [13] a las tres dimensiones ($MSR_{3D}$).
- Proponer y desarrollar una medida de evaluación de triclusters inspirada en la recta de mínimos cuadrados para una serie de puntos ($LSL$).
- Proponer y desarrollar una medida de evaluación de triclusters basada en un enfoque gráfico de los mismos ($MSL$).
- Comparación de efectividad de las medidas de evaluación propuestas.
- Definir los aspectos fundamentales para evaluar la calidad de un tricluster y proponer una medida a tal efecto. De esta forma, se consigue aglutinar dichos aspectos que en la actualidad se tienen en cuenta por separado ($TRIQ$).

- Probar la efectividad de las metodologías en datasets biológicos y sintéticos.
- Probar la efectividad de las metodologías en campos no biológicos.
- Integrar todos los aspectos anteriores en una herramienta accesible y de fácil uso por el experto biológico que sirva de apoyo en la toma de decisiones (metodología *TrLab*).

## 1.4. Estructura

A continuación se detalla la estructura en secciones del presente documento:

**Parte I**

**Capítulo 1**
En este capítulo se hace una descripción del problema en el que se enmarca esta investigación (Sección 1.1) así como la justificación de la realización del mismo (Sección 1.2) y los objetivos a completar (Sección 1.3).

**Capítulo 2**
Con este capitulo se pretende contextualizar y definir el escenario científico de esta investigación. Se hace hincapié en los campos de la Minería de Datos (Sección 2.1) y la Bioinformática (Sección 2.2) como punto de referencia de las técnicas y metodologías estudiadas y desarrolladas. Se analizan los conceptos de microarray (Sección 2.3) y de datos de estudios sísmicos (Sección 2.4) como recursos fundamentales del modelo desarrollado y se define la técnica del Triclustering (Sección 2.5). Además, se presentan otros conceptos que conforman el núcleo de esta investigación

como los Algoritmos Genéticos (Sección 2.6), el proyecto *Gene Ontology* (Sección 2.7), y los índices de correlación (Sección 2.8).

**Capítulo 3**

En este capítulo se pretende ofrecer un resumen de todos los resultados aportados en esta investigación y que forman parte de la metodología *TrLab*. En primer lugar se describe de una manera global el algoritmo *TriGen* (Sección 3.1). Seguidamente, se detallan las distintas funciones de evaluación desarrolladas: $MSR_{3D}$ (Sección 3.2), $LSL$ (Sección 3.4) y $MSL$ (Sección 3.5). También, se realiza una descripción de la medida de calidad $TRIQ$ (Sección 3.6) y una explicación general de la aplicación al análisis de datos sísmicos (Sección 3.7). El capítulo concluye con una discusión conjunta de los resultados obtenidos (Sección 3.8).

**Parte II**

**Capítulo 4**

Se presentan todas las publicaciones en revistas incluidas en el *Journal Citation Reports* de *Thomsom-Reuters* fruto de esta investigación y que forman parte del compendio de publicaciones de esta tesis doctoral. Por cada publicación se indica información sobre editorial, campo de investigación y ranking *JCR*.

**Capítulo 5**

Contiene el resto de publicaciones, no *JCR*, que también son fruto de este trabajo. Éstas están dispuestas en congresos nacionales (Sección 5.1), congresos internacionales (Sección 5.2) y *Lecture Notes in Computer Science* (Sección 5.3).

**Parte III**

**Capítulo 6**
Detalle de todas las conclusiones obtenidas de esta investigación.

**Capítulo 7**
Próximas tareas realizables y futuras líneas de investigación en el contexto de esta investigación.

**Apéndice A**
Currículum vítae del autor de esta investigación.

# Capítulo 2

# Contexto de Investigación

El objeto de este capítulo es contextualizar los conceptos y ámbitos en los que se mueve esta investigación. Primero se introduce el campo de la Minería de Datos (Sección 2.1), marco de referencia fundamental en el que se encuadra dicha tesis, así como el campo de la Bioinformática (Sección 2.2) que también es clave en la constitución de este trabajo. Seguidamente se analizarán los dos recursos esenciales en los que se ha basado este trabajo de investigación y que conforman la entrada del modelo diseñado: los datos de microarray (Sección 2.3) y los datos de análisis de seísmos (Sección 2.4). Por último se detallan el catálogo de técnicas empleadas y desarrolladas para llevar a cabo los objetivos de esta investigación empezando por la principal, el Triclustering (Sección 2.5), como técnica global y siguiendo con otras técnicas y metodologías empleadas como los Algoritmos Genéticos (2.6), el proyecto *Gene Ontology* (Sección 2.7) y los coeficientes de correlación (Sección 2.8).

## 2.1. Minería de Datos

La Minería de Datos (Data Mining) es entendida de manera general como el conjunto de técnicas computacionales diseñadas para encontrar, información oculta, válida y útil de entre la vasta cantidad de datos que fluye de manera continuada y a diario en nuestro mundo. Es una especialidad interdisciplinaria que comprende un alto número de campos de investigación tales como la estadística, la recuperación de la información, la inteligencia artificial, la gestión de bases de datos, el soporte para la toma de decisiones, el reconocimiento de patrones, la visualización de datos o el aprendizaje automático; con lo cual, se presta a ser definida de diferentes maneras.

Dos de las más aceptadas en la comunidad científica (aunque no liberadas de debate y controversia) son, por una parte, el considerar el término *Minería de Datos* como un sub-proceso de una disciplina de más alto nivel conocida como *KDD* (*Knwoledge Discovery from Data*) o, de otro modo, el considerar ambas disciplinas como equivalentes. En cualquier caso, ya se considere Minería de Datos como un sub-proceso del *KDD* o como su igual, la esencia de ambos términos es la de agrupar todas las técnicas en el ámbito de la Ingeniería Informática que se encargan de extraer información oculta, válida y útil de conjuntos de datos heterogéneos de gran tamaño. Por lo tanto, cuando se hace referencia a términos como Minería de Datos, *KDD*, extracción de conocimiento, análisis de patrones de datos, arqueología de datos, etc, estamos refiriéndonos, en esencia, al mismo concepto.

En Fig. 2.1 podemos observar como el proceso global del *KDD*, entendido como extracción de conocimiento, está formado por varios sub-procesos y, como antes se indicó, la Minería de Datos conforma el núcleo del procedimiento global en cuanto a la extracción de conocimiento se refiere y, además, es la disciplina que surte de técnicas necesarias para tal efecto.

El motivo de distinguir entre *KDD* y *Minería de Datos* en este trabajo es el de separar las técnicas de *Pre-procesado de Datos* y *Representación de conocimiento*, consideradas tareas con un campo de trabajo muy amplio, de la extracción de conocimiento propiamente dicha.

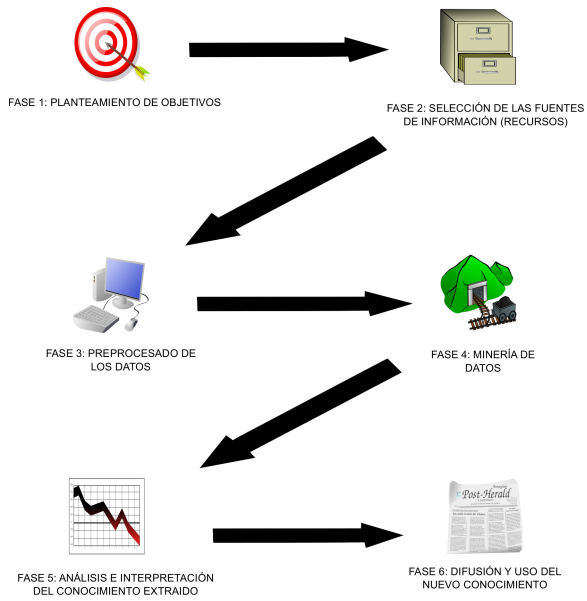



Figura 2.1: Proceso del *KDD*

Las diferentes etapas del *KDD* se definen como sigue [32]:

1. Planteamiento de objetivos.

   Como en todo proyecto de Ingeniería Informática que se precie, el planteamiento de objetivos, establecimiento de cotas del problema así como la planificación del mismo, supone un paso importante y crítico que influirá en el desarrollo del proceso. El *KDD* al ser un proceso ingenieril tiene a este sub-proceso como primer paso de su procedimiento global.

2. Selección de las fuentes de información.

   Búsqueda de los datos que nos servirán de recurso para extraer conocimiento.

3. Pre-procesado de datos.

   Preparar los datos para asegurar la efectividad de la extracción de conocimiento. Los procedimientos para este paso son la *Limpieza de los Datos* para eliminar ruido e inconsistencias, *Integración de Datos* para homogeneizar y unir distintas fuentes o recursos en una sola, la *Selección* donde los datos relevantes para la tarea de análisis son recuperados de las fuentes de información y *Transformación de Datos* donde son transformados y consolidados en las formas apropiadas para el análisis mediante operaciones de sumatorio, agregación y reducción.

4. Minería de Datos.

   Extracción de conocimiento propiamente dicha usando diferentes técnicas.

5. Análisis e interpretación del conocimiento extraído.

   El conocimiento extraído se analiza e interpreta para su uso práctico posterior. Importante en este paso también es la evaluación de la calidad del modelo resultante.

6. Difusión y uso del conocimiento extraído.

   Técnicas de visualización y representación del conocimiento entran en juego para difundir y usar el conocimiento obtenido.

### 2.1.1. Datos

Las técnicas de Minería de Datos son susceptibles de ser aplicadas a datos de todo tipo de origen y naturaleza. Los orígenes de datos más comunes son las bases de datos, los almacenes de datos, y los datos transaccionales así como otras disposiciones como streams, datos secuenciales u ordenados, grafos o redes, datos espaciales, datos de textos, datos multimedia y de la *World Wide Web*. Se puede afirmar que esta disciplina procura abarcar todos los tipos de datos que existen según van emergiendo además de observar que todos los modelos o conocimientos extraídos de los distintos tipos de datos contienen los adjetivos de oculto, válido y útil.

El objeto de aplicar técnicas de Minería de Datos a bases de datos relacionales es obtener más información que la que nos proporciona una consulta, es decir, pretendemos encontrar tendencias o patrones en los datos. Por ejemplo para una base de datos de un comercio, estas técnicas pueden analizar datos de clientes para predecir las futuras ventas en función de la edad y las compras previas así como detectar desviaciones, esto es, productos con ventas lejos de las expectativas formadas en años anteriores.

Los almacenes de datos son repositorios de información recolectada de múltiples fuentes localizadas físicamente en distintos lugares y almacenadas bajo un esquema único. Éstos son generalmente modelados como una estructura multidimensional conocida como *Cubo de Datos* en el que cada dimensión corresponde con un atributo o con un conjunto de atributos del esquema donde, para cada celda, se almacena el valor de alguna medida de resumen como un contador o una suma. De esta forma, se proporciona una vista multidimensional de los datos y se habilita el acceso rápido a los datos resumidos. Para poder proporcionar esta funcionalidad de multidimensionalidad y resumen, los almacenes de datos contienen un inherente soporte para operaciones *OLAP* (Online Analytical Processing Operations); estas operaciones realizan accesos a los datos a diferentes niveles de abstracción. Las técnicas de *Minería de Datos* permiten la exploración de múltiples combinaciones de dimensiones variando el nivel de granularidad y, por lo tanto,

proporcionan el descubrimiento de interesantes patrones o tendencias que representan conocimiento válido, oculto y útil.

Los datos transaccionales son los más sencillos en cuanto a estructura o esquema. Un tipo de dato transaccional es un conjunto de transacciones. Se entiende como transacción cada uno de los registros de una base de datos (una compra de un cliente, una reserva de un vuelo, etc) y por lo general incluye un identificador único y una lista de ítems que la componen (como por ejemplo, los productos adquiridos en una compra). De manera genérica, estos datos suelen ir presentados en ficheros y en forma de tabla de modo que cada fila de dicha tabla representa un registro de dicha transacción y cada columna representa un atributo de la misma; por ejemplo, para el caso de un comercio, el conjunto de datos transaccionales de todas las ventas del mismo estaría representado en un fichero donde cada línea representa una venta y cada columna de la linea, distinguidas por un carácter separador (comúnmente la coma o el punto y coma), se corresponden con cada uno de los valores de los atributos de dicha venta. Un gran número de técnicas de Minería de Datos usan estas fuentes como recursos para extraer conocimiento orientados a encontrar patrones frecuentes, de comportamiento u otros. En muchos trabajos realizados en el campo de la Bioinformática (Sección 2.2), Bussines Intelligence y Toma de Decisiones se hace uso de la Minería de Datos aplicada a este tipo de recurso. En la presente investigación, se analiza una clase particular de estos datos como recurso de investigación: los Datos Longitudinales (Sección 2.3).

Además de las ya expuestas, existen una gran variedad de fuentes con formas y estructuras versátiles de las que la Minería de Datos puede extraer conocimiento. Estos tipos de datos pueden provenir de muchos tipos de aplicaciones y orígenes, por ejemplo, datos secuenciales como registros históricos, series temporales, secuencias biológicas, streams de vídeo, audio o sensores, datos espaciales como mapas, datos de diseño en ingeniería como planos de edificios, diseño de software, componentes de sistemas o circuitos integrados y datos del *World Wide Web* como paginas web o redes sociales.

Los tipos de conocimiento que las técnicas de Minería de Datos pueden extraer son tan numerosos como las fuentes existentes, estos son, por ejemplo, los bancos de tendencias de compras, los modelos para detectar acciones no permitidas en sistemas informáticos, la clasificación de opiniones, la caracterización de páginas web, la recuperación y clasificación de textos científicos en base a temas, etc. Podríamos decir que, potencialmente, el límite de esta disciplina lo establecen los recursos (los datos) en lugar de las técnicas.

### 2.1.2. Técnicas

Las técnicas del campo de la Minería de Datos se clasifican de manera general en descriptivas o predictivas [32]. Las primeras, como su propio nombre indica, describen los datos mientras las segundas realizan predicciones de los mismos o, dicho de otra manera, las descriptivas caracterizan propiedades del conjunto de datos objetivo mientras que las predictivas inducen cierto conocimiento para realizar predicciones sobre los datos objetivo.

Entre todo el catálogo de técnicas, destacan por su carácter representativo dentro del campo: el análisis de outliers, la descripción de conceptos, el descubrimiento de patrones frecuentes, asociaciones y correlaciones, el análisis predictivo, la regresión y el Clustering.

Los *outliers* son objetos o items contenidos en los datos de entrada que no concuerdan con el comportamiento o modelo del mismo. Existen multitud de metodologías para la detección y eliminación de outliers y así evitar ruido y excepciones, sin embargo, en algunas aplicaciones estos eventos son más interesantes que los propios datos fuente. El análisis de outliers o minería de anomalías se encarga de detectar estos eventos de interés.

Los datos susceptibles de análisis pueden estar asociados a clases o conceptos (por ejemplo en del caso del comercio, pueden existir los conceptos de "gran comprador" y "comprador moderado" asociado a los clientes). Puede

ser útil describir estos conceptos en términos concisos, resumidos y precisos. Esta técnica de descripción de conceptos se puede desarrollar en tres vías: *Caracterización de Datos* en la que se resumen los datos del concepto bajo estudio, *Discriminación de Datos* en la que se compara el concepto objetivo con un conjunto de conceptos comparativos, o ambos enfoques trabajando conjuntamente. Existen varios métodos para realizar estas tareas de caracterización y discriminación tales como resúmenes basados en medidas estadísticas, operaciones *OLAP* sobre los cubos de datos, métodos de inducción orientados al atributo, etc. Los resultados de la aplicación de estas técnicas son presentados en varios formatos: gráficas circulares, gráficos de barras, curvas, cubos y tablas multidimensionales, etc.

Obtener *patrones frecuentes* permite descubrir asociaciones y correlaciones interesantes dentro de los datos de entrada. Entre los múltiples tipos de variantes de patrones frecuentes existentes son destacables los patrones que informan de un conjunto de objetos que suelen aparecer juntos en un conjunto de datos transaccional (por ejemplo, en un supermercado, los productos leche y pan suelen ir juntos en la misma cesta de la compra de muchos clientes), los patrones de subsecuencias que proveen de un conjunto de objetos que suelen aparecer en el conjunto de datos en un orden concreto (por ejemplo, los clientes de una tienda de tecnología que tienden a comprar primero un ordenador portátil, seguido de una cámara digital y una tarjeta de memoria), los patrones de subestructuras que indican disposiciones que suelen repetirse dentro de una estructura global como pueden ser grafos, árboles, etc (por ejemplo, patrones de estructuras tridimensionales que se repiten dentro de una proteína) y los patrones de comportamiento que proporcionan un conjunto de elementos dentro de los recursos que tienen un comportamiento similar acorde a la naturaleza dichos recursos (por ejemplo, de las señales de comunicación que recibe una antena, agrupar las que tengan un comportamiento similar en base a su longitud de onda).

La técnicas de *Clasificación* consisten en encontrar un modelo (o función) que describa y distinga los conceptos o clases de los datos fuente. El modelo es obtenido en base al análisis de un conjunto de *datos de entre-*

*namiento* (esto es, datos cuyo concepto o clase es conocido). El modelo se usa para predecir la clase (o clasificar) los objetos que no tienen clasificación alguna. Para obtener el modelo resultante de la aplicación, existen varios métodos. Entre los más usados podemos destacar los árboles de decisión, las reglas de asociación, las redes neuronales, las redes bayesianas, las máquinas de vector soporte y el algoritmo del vecino más próximo.

Mientras que la *Clasificación* predice la categoría de objetos discretos, la *regresión* modela funciones de valores continuos. La *Regresión* es una metodología estadística que es mayoritariamente usada para predicción numérica. Además, abarca la identificación de la distribución de tendencias en los datos procesados.

Al contrario que la *Regresión* o la *Clasificación* que analizan conjuntos de datos con etiquetas (conjunto de entrenamiento, aprendizaje supervisado), el *Clustering* analiza los datos de entrada sin atender a dicha etiqueta (aprendizaje no supervisado). En muchos casos, no se puede determinar un conjunto de entrenamiento con las etiquetas de clase. Para ello, el Clustering puede ser usado para generarlas. De manera global, estas técnicas separan en grupos o *clusters* el conjunto completo de los datos de entrada basándose en el principio de maximizar la similitud entre los elementos de un grupo y minimizando la similitud entre los elementos de grupos distintos. El *Clustering* puede facilitar la formación taxonómica, o lo que es lo mismo, la organización de observaciones en clases jerárquicas que agrupan eventos similares.

## 2.2. Bioinformática

En las últimas décadas, los avances en la biología molecular y el equipamiento disponible para la investigación en este campo, han permitido la rápida secuenciación de grandes porciones de genomas de diversas especies.

En la actualidad, muchos genomas procariotas, tales como la bacteria *Escherichia coli* y muchos eucariotas ya han sido secuenciados por completo. El proyecto Genoma Humano [15], diseñado con el fin de secuenciar los 46 cromosomas del ser humano, también está terminado. Las bases de datos de secuencias más populares, como *GenBank* [7] y *EMBL* [37], están creciendo de forma exponencial. Esta gran cantidad de información necesita de un alto nivel de organización, indexado y almacenamiento de las secuencias. Es por ello que la Ingeniería Informática ha sido aplicada a la Biología para producir un nuevo campo de investigación llamado Bioinformática que permita ayudar a esta organización [1].

El término Bioinformática ha sido adoptado por varias disciplinas diferentes. En su sentido más amplio, puede considerarse que el término significa tecnología de la información aplicada a la gestión y análisis de datos biológicos. Esto tiene implicaciones en diversas áreas, desde la inteligencia artificial y la robótica al análisis de genomas.

En el contexto de los proyectos genoma, el término se aplicó originalmente a la manipulación computacional y al análisis de datos de secuencias biológicas (ADN o proteínas), sin embargo, a la vista de la rápida y creciente acumulación de estructuras de proteínas disponibles, el término actualmente abarca numerosos campos como el análisis y predicción de estructuras proteínicas tridimensionales, cálculo de propiedades fisico-químicas de las proteínas, clasificación evolutiva de las proteínas, análisis de expresión genética, etc.

### 2.2.1. Objetivos y Áreas de trabajo

La Bioinformática acepta un enfoque desde la perspectiva de la Ingeniería Informática como campo de investigación interdisciplinar en el que confluyen varias especialidades como las Bases de Datos, la Ingeniería del Software o la Minería de Datos en las que destacan las tareas de diseño, acceso y análisis de bases de datos biológicas, el análisis de secuencias y estructuras moleculares, la visualización y predicción de estructuras 3D de macromoléculas y el reconocimiento de patrones de comportamiento genéticos.

Desde este punto de vista, la Bioinformática engloba el estudio y solución de problemas relacionados con datos de origen biológico. Éstos constituyen un recurso muy amplio y pueden ser clasificados en tres campos de investigación que agrupan distintas tareas y problemas abiertos [14]: el análisis de secuencias de ADN/ARN, el análisis de secuencias y estructuras proteicas y el análisis de genomas completos.

Uno de los elementos principales en la Bioinformática es el campo del análisis de secuencias de ADN/ARN. El *Ácido Desoxirribonucleico* (ADN) es una macro molécula en forma de cadena plegada en doble hélice en el que cada eslabón de la misma lo constituyen un par de nucleótidos; desde el punto de vista funcional el ADN contiene instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos conocidos y algunos virus, y es responsable de su transmisión hereditaria. Del mismo modo, el *Ácido Ribonucleico* (ARN) es una macro molécula en forma de cadena simple en el que cada nodo de la misma lo forma un nucleótido y su función es la de dirigir las etapas intermedias de la síntesis proteica. Una vista de la morfología molecular de ambas cadenas puede apreciarse en Fig. 2.2.

De forma general y muy simplificada, se puede decir que el ADN contiene las instrucciones para la construcción de proteínas y el ARN es el encargado de ejecutarlas. Tanto el ADN como el ARN constituyen un lenguaje de bajo nivel con todo lo que ello supone: símbolos de codificación,

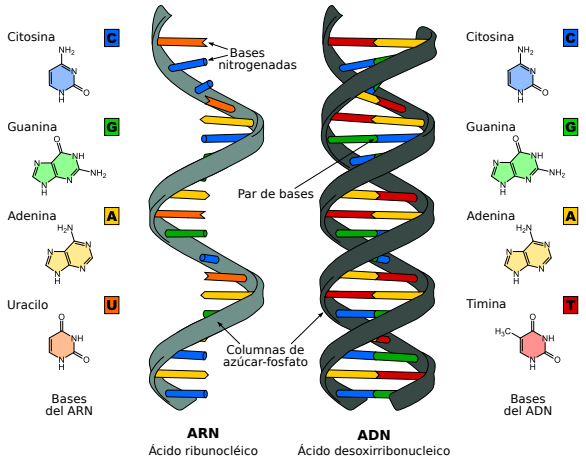



Figura 2.2: ARN y ADN

sintaxis e interpretación. Ambos lenguajes tienen 4 símbolos de codificación o bases nitrogenadas (parte central de los eslabones o nucleótidos): *Citosina* (C), *Adenina* (A), *Guanina* (G) para ambos, *Timina* (T) para el ADN y *Uracilo* (U) para el ARN. Como sintáxis básica, cada base nitrogenada tanto en la doble hélice del ADN como en el proceso de transcripción en el ARN va asociada a su complementaria de tal forma que la *Adenina* se complementa con la *Timina* o *Uracilo* y la *Citosina* con la *Guanina*. La interpretación básica es que cada triplete de bases nitrogenádas corresponde con un aminoácido. Los elementos que constituyen el análisis de secuencias de ADN/ARN se pueden observar de manera global en Fig. 2.3.

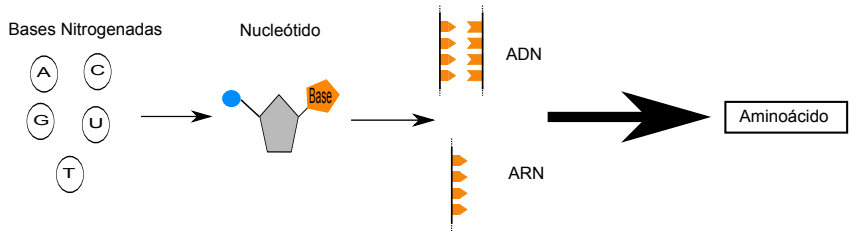



Figura 2.3: Elementos del análisis de secuencias de ADN/ARN

De este campo, surgen multitud de tareas y problemas a abordar debido a la alta complejidad del sistema biológico de los seres vivos, entre los más destacados, se encuentran [14]:

1. Recuperación de secuencias de ADN en bases de datos biológicas.
2. Cálculo de composición de nucleótidos.
3. Identificación de zonas de restricción.
4. Diseño de cebadores para reacciones en cadena de la polimerasa.
5. Identificación de marcos abiertos de lectura.
6. Predicción de elementos de ADN/ARN de estructura secundaria.
7. Búsqueda de repeticiones.
8. Cálculo del alineamiento óptimo entre dos o más secuencias de ADN.
9. Búsqueda de zonas polimórficas en genes.
10. Ensamblaje de fragmentos de secuencias.

11. Análisis de la composición de una secuencia de ADN.
    *a*) Establecer el contenido de Guanina (G) y Citosina (C).
    *b*) Conteo de palabras.
    *c*) Encontrar repeticiones internas..
    *d*) Encontrar regiones de codificación de proteínas.
    *e*) Encontrar exones internos en secuencias genómicas.
12. Ensamblaje de fragmentos de secuencias.
13. Predicción y modelado de estructuras secundarias de ARN.
14. Búsqueda de similitudes en bases de datos de secuencias de ADN.
15. Comparación de dos secuencias de ADN.
16. Comparación de múltiples secuencias de ADN.

En el campo del análisis de secuencias y estructuras proteicas, la proteína y sus distintas configuraciones estructurales es el objeto de estudio fundamental. Una proteína es una macro molécula constituida por cadenas de moléculas orgánicas con un grupo amino y un grupo carboxilo unidos a un carbono central llamadas aminoácidos. Las proteínas ocupan un lugar de máxima importancia entre las moléculas constituyentes de los seres vivos. Prácticamente todos los procesos biológicos dependen de la presencia o la actividad de este tipo de moléculas. Al ser una molécula compleja, la proteína está constituida por cuatro niveles estructurales que están íntimamente ligados con su función en el organismo del ser vivo correspondiente (Fig. 2.4).

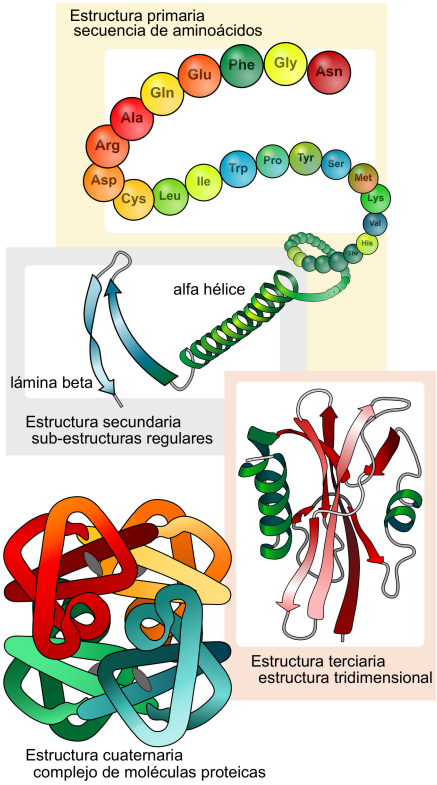



Figura 2.4: Estructuras proteicas

La relación de las proteínas con el nivel ADN/ARN se establece gracias a uno de los descubrimientos científicos más importantes de la historia: el *código genético*, que establece la relación directa entre los tripletes de nucleótidos y los aminoácidos que sintetizan. Como podemos ver en Fig. 2.5 cada combinación de tres nucleótidos (parte interior de la rueda) sintetiza un aminoácido concreto o las órdenes de empezar la transcripción («start») y terminar la transcripción («stop») (perímetro de la rueda). De este modo, el primer paso en la síntesis de una proteína sería activar el triplete de

«start» (*Adenina - Uracilo - Guanina*), que sintetiza el aminoácido *Metionina*, establecer las secuencias de tripletes para los siguientes aminoácidos y activar alguna de las tres secuencias de «stop» (*Uracilo - Adenina - Adenina*, *Uracilo - Adenina - Guanina* o *Uracilo - Guanina - Adenina*).

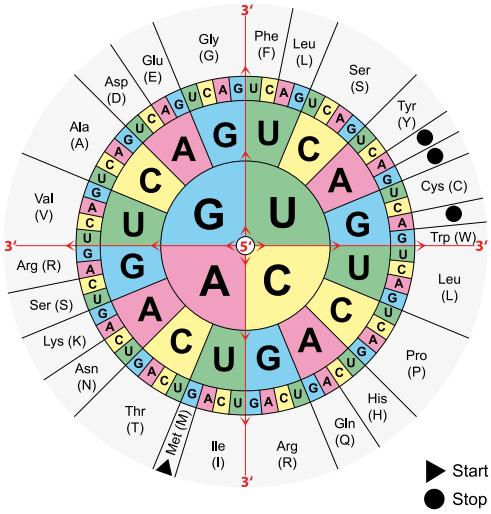



Figura 2.5: Código Genético

Los aminoácidos son las unidades mínimas de información dentro de las proteínas. Una proteína, en origen, es una combinación secuencial de estas unidades mínimas de información que puede adoptar distintas estructuras en el espacio (Fig. 2.4). De esta forma, y al igual que en el ADN/ARN, podemos interpretar el nivel protéico como un lenguaje de 20 símbolos (20 aminoácidos existentes en la naturaleza) en el que cada proteína es una combinación secuencial de los mismos. Los elementos que constituyen análisis de secuencias y estructuras proteicas se pueden observar de manera global en Fig. 2.6.

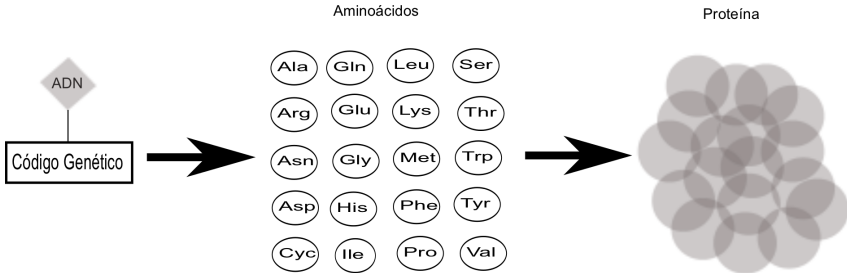



Figura 2.6: Elementos del análisis de secuencias y estructuras proteicas

Entre los retos bioinformáticos más destacados de este campo se encuentran [14]:

1. Recuperación de secuencias de proteínas en bases de datos biológicas.
2. Cálculo de propiedades de los aminoácidos: composición, peso molecular, etc.
3. Cálculo del nivel de hidrofobia y hidrofilia de una proteína, predicción de antígenos.
4. Predicción de elementos de estructura secundaria.
5. Predicción del dominio de organización de la proteína.
6. Visualización de la estructura 3D de la proteína.
7. Predicción de la estructura 3D de la proteína a partir de su secuencia de aminoácidos.
8. Búsqueda de proteínas que comparten una secuencia similar de aminoácidos.
9. Clasificación de proteínas en familias.
10. Búsqueda del alineamiento óptimo entre dos o más proteínas.

11. Búsqueda de relaciones evolutivas entre proteínas; dibujo de árbol genealógico proteico.
12. Predecir las principales propiedades fisico-químicas de una proteína.
13. Análisis de la estructura primaria.
    *a*) Búsqueda de segmentos posicionados en la membrana.
    *b*) Búsqueda de regiones coiled-coil.
14. Predicción de modificaciones post-traducción.
    *a*) Búsqueda de patrones *PROSITE*.
    *b*) Búsqueda de dominios conocidos.
    *c*) Búsqueda de nuevos dominios.
15. Predicción de la estructura secundaria de una proteína.
16. Predicción características estructurales de una proteína.
17. Predicción de la estructura terciaria de una proteína.
18. Búsqueda de similitudes en bases de datos de secuencias de proteínas.
19. Comparación de dos secuencias de proteínicas.
20. Comparación de múltiples secuencias de proteínicas.

El tercer campo de investigación es el análisis de genomas completos. El genoma es la totalidad de la información genética que posee un organismo en particular. Desde el punto de vista estructural, el genoma se organiza en unidades llamadas cromosomas que, a su vez, se agrupan en otras unidades mínimas, los genes. Un cromosoma es una estructura molecular compleja formada por un conjunto determinado de genes que realizan una función biológica concreta en el ser vivo. Un gen es una secuencia ordenada de nucleótidos en la molécula de ADN que contiene la información necesaria

para la síntesis de una proteína con función celular específica. El genoma humano contiene 46 cromosomas y un número de genes del orden de 25.000. Desde el punto de vista de la ingeniería podemos decir que el genoma es el repositorio completo de recursos para construir funciones biológicas de un ser vivo; esto es, en el genoma encontraremos todos los genes encargados de cada función biológica organizados por cromosomas (Fig. 2.7). De entre los problemas abordables más destacados en Bioinformática para este campo, destacan [14]:

1. Búsqueda de genomas disponibles.
2. Análisis de secuencias en relación a genomas específicos.
3. Análisis de expresión genética.
4. Representación gráfica de genomas.
5. Análisis de genomas microbianos.
6. Análisis de genomas eucarióticos.
7. Búsqueda de genes homólogos.
8. Búsqueda de repeticiones.

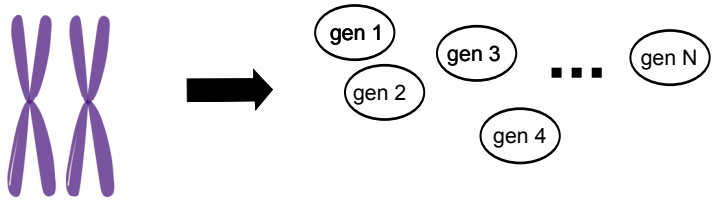



Figura 2.7: Elementos del análisis de genomas completos

En este análisis de los objetivos y áreas de trabajo sólo se ha reflejado algunas de las disciplinas más recurrentes que tienen cabida bajo el término Bioinformática. No es objetivo de esta investigación el hacer un análisis exhaustivo de todas las disciplinas. Cabe destacar que hay otros muchos logros en investigación dentro de la Bioinformática que hacen que dichas disciplinas vayan aumentando en número.

## 2.3. Microarrays

Los complejos procesos moleculares de los sistemas biológicos han logrado ser estudiados gracias a los progresos en biología molecular junto con el avance de las técnicas computacionales y el hardware [22]. Dicho hecho ha supuesto una revolución tecnológica que se ha traducido en el aumento exponencial de la cantidad de datos disponibles. En particular, los chips de alta densidad de nucleótidos o ADN complementario desarrollados en los años 90, conocidos como microarrays, han revolucionado la investigación biológica por su capacidad para monitorizar los cambios en la concentración de ARN en miles de genes simultáneamente [9] mientras que los métodos tradicionales podían únicamente observar un gen cada vez.

El funcionamiento de los microarrays se basa en hacer valer la capacidad de las moléculas de ARN mensajero (encargado de transportar la información sobre la secuencia de aminoácidos de la proteína al ribosoma) para unirse específicamente con el ADN que lo produjo. Partiendo de un biochip que contenga una gran cantidad de muestras de ADN el experto científico puede determinar los niveles de expresión genética de miles de genes de una célula mediante la medición de la cantidad de ARN mensajero ligado a cada sección del biochip. Cada una de estas medidas es calculada de manera precisa por un ordenador produciendo un perfil o colección de los niveles de expresión genética de la célula bajo estudio. La producción de estas colecciones de niveles de expresión genética ha dado pie a la identificación y el estudio de los patrones de expresión genética que subyacen a la fisiología celular, esto es, podemos ver qué genes se encuentran activa-

dos (o reprimidos) bajo distintas condiciones o ante la presencia de agentes externos. Este campo de investigación es conocido como análisis de datos de expresión genética [8].

Una práctica común en el análisis de los datos de expresión génica consiste en aplicar técnicas de agrupamiento. Estos de grupos de genes muestran patrones similares de expresión y son interesantes porque se considera que los genes con patrones de comportamiento similares pueden estar involucrados en procesos de regulación similares [72]. Aunque, en teoría, no hay un gran paso de la correlación a la similitud funcional de los genes, varios artículos indican que esta relación existe [19].

### 2.3.1. Aplicaciones

Físicamente, un microarray o biochip (Fig. 2.8) es una colección de pequeños fragmentos de genes unidos a la superficie de pequeños cristales. En ellos, se integran decenas de miles de fragmentos de material genético de secuencia conocida y de diferente tamaño ordenados sobre un sustrato sólido de manera que forman una matriz de secuencias en dos dimensiones. Dichos fragmentos se denominan sondas.

Las muestras a analizar son marcadas por diversos métodos (enzimáticos, fluorescentes, etc.) incubándose posteriormente sobre la matriz de sondas y produciéndose una hibridación entre las secuencias homólogas, es decir, sólo las cadenas complementarias a las del chip son hibridadas. Después de la hibridación entre las secuencias del microarray y la muestra marcada con fluorescencia, los chips son leídos en un escáner originándose un patrón de luz característico y una cuantificación de la intensidad de hibridación de cada punto. Los datos obtenidos son interpretados mediante un ordenador lo que permite una identificación y cuantificación del ADN o ARN presente en la muestra.

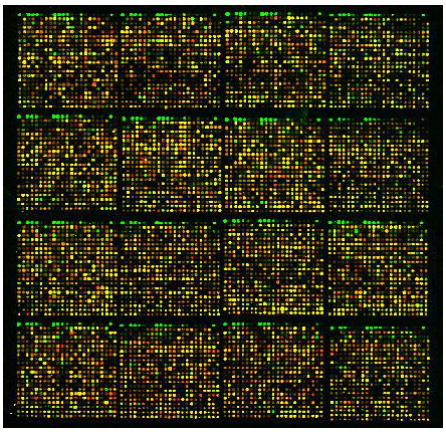


Figura 2.8: Microarray

Una de las aplicaciones más extendidas de esta tecnología es la comparación de genes de individuos sanos con enfermos. Por ejemplo, si tenemos muestras de pacientes con cáncer de colon, y muestras de pacientes sanos, podemos comparar el nivel numérico de expresión de los genes en ambas muestras, extraídos en función de la intensidad lumínica del microarray en esas celdas, identificar los genes que se comportan distinto en el paciente sano y en el enfermo y estudiarlos más en profundidad pues potencialmente pueden estar relacionados con el cáncer de colon.

En Fig. 2.9 podemos ver una imagen real de dos microarrays que conforman otro ejemplo de este corte: el microarray de la izquierda corresponde a un ratón enfermo mientras que el de la derecha es de un ratón sano. Se puede observar que hay un par de genes que, bajo una misma condición experimental, tienen diferentes niveles de expresión para el enfermo y el sano. Este hecho se contempla en los colores de los genes seleccionados, el del ratón sano es diferente del enfermo, de esta forma, el color en un microarray corresponde con el nivel de expresión.

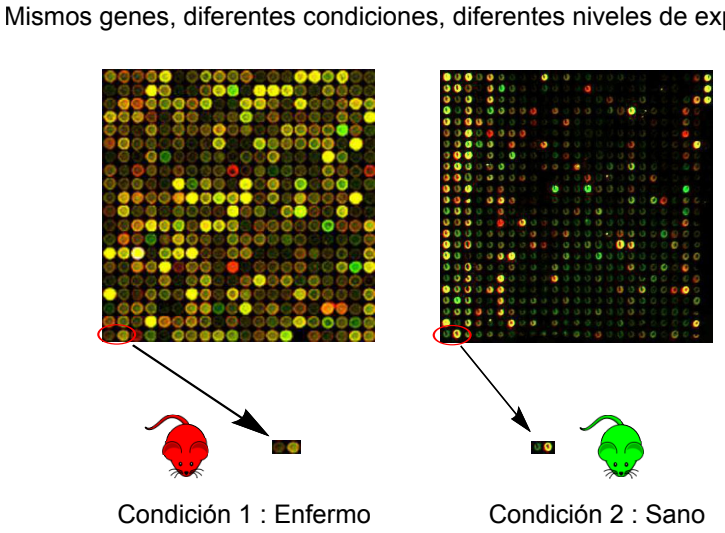



Figura 2.9: Experimento con microarray

Para poder aplicar técnicas computacionales sobre datos de microarray es necesario usar un equivalente digital a los biochips sintetizados en el laboratorio. Dicho equivalente digital no es más que una correspondencia entre el nivel de color de una celda con un valor numérico. Podemos observar un ejemplo reducido del mismo en Fig. 2.10. En ella, vemos como cada fila representa a un marcador genético mientras que cada columna representa a un gen. Los valores numéricos se corresponden con los colores del microarray físico.

Los datos digitales de microarray se pueden interpretar como una colección de niveles de expresión genética para una particular condición experimental. En esta colección de niveles, cada celda es un color que representa un nivel de expresión genética bajo una condición experimental concreta.

Para el procesamiento computacional, se agrupan diferentes experimentos de microarray para obtener una tabla indexada por gen (filas) y condiciones experimentales (columnas); cada una de las celdas de esta tabla

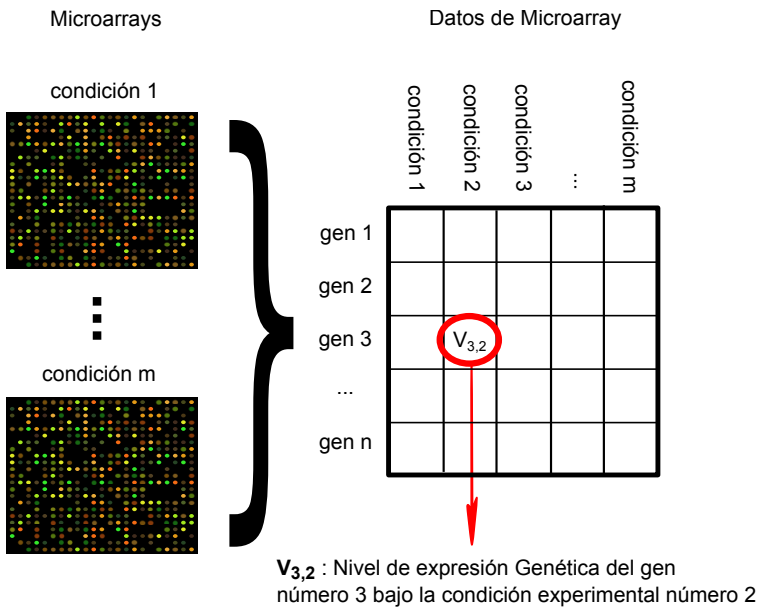



Figura 2.10: Datos de microarray

contendrá un número real que representa el nivel de expresión genética. Como se puede observar en Fig. 2.10, la celda de fila 3 columna 2 es un número real que determina el nivel de expresión del gen número 3 bajo la condición experimental número 2.

### 2.3.2. Experimentos temporales

En la naturaleza existen numerosos procesos cuyo desarrollo cambia según avance del tiempo. Reacciones de productos químicos, fenómenos geológicos como los terremotos, procesos biológicos como los ciclos celulares o los procesos fisiológicos de los seres vivos son un pequeño subconjunto de ejemplos de entre el amplio abanico que existe en nuestra realidad.

Los datos temporales extraíbles de estos procesos se definen de manera general como datos que varían en el tiempo o como datos longitudinales [81].

Más exhaustivamente pueden ser denominados como un conjunto de $I$ instancias y $A$ atributos, cuyos valores varían a lo largo de $T$ puntos de tiempo. Adoptando una visión geométrica, tendremos un cubo de $I$x$A$x$T$ celdas en la que cada celda de fila $i \epsilon I$, columna $a \epsilon A$ y ancho $t \epsilon T$ representa al valor que tiene la instancia $i$ para el atributo $a$ en el instante de tiempo $t$. Este concepto se puede observar en Fig. 2.11, si escogemos una celda, por ejemplo la $V22$, vemos que su valor será distinto en cada punto de tiempo (tiempo 1, 2, ..., $T$).

```
Tiempo 1    Atributo 1 Atributo 2  ... Atributo A
Instancia 1    V11          V12    ...    V1A
Instancia 2    V21          V22    ...    V2A
    ...         ...          ...    ...    ...
Instancia I    VI1          VI2    ...    VIA

Tiempo 2    Atributo 1 Atributo 2  ... Atributo A
Instancia 1    V11          V12    ...    V1A
Instancia 2    V21          V22    ...    V2A
    ...         ...          ...    ...    ...
Instancia I    VI1          VI2    ...    VIA
.
.
.
Tiempo T    Atributo 1 Atributo 2  ... Atributo A
Instancia 1    V11          V12    ...    V1A
Instancia 2    V21          V22    ...    V2A
    ...         ...          ...    ...    ...
Instancia I    VI1          VI2    ...    VIA
```

Figura 2.11: Datos longitudinales

Así pues, un experimento de microarray temporal se compone de la síntesis en laboratorio de varios microarrays realizados en instantes de tiempo diferentes. De esta forma, como resultado de estos experimentos tendremos tantos biochips como puntos de tiempo sean considerados. En Fig. 2.12 podemos ver un ejemplo real de un experimento microarray temporal. En ella, podemos observar un microarray por cada fase del ciclo celular de una célula (fases $G_0$, $G_1$, $S$, $G_2$ y $M$). Cada microarray se corresponde con un punto de tiempo, por lo tanto, se puede comparar el nivel de expresión de un gen en cada punto de tiempo o fase del ciclo celular. En el ejemplo de Fig. 2.12 vemos que un gen concreto tiene diferentes niveles de expresión o colores bajo la misma condición experimental en diferentes puntos de tiempo.

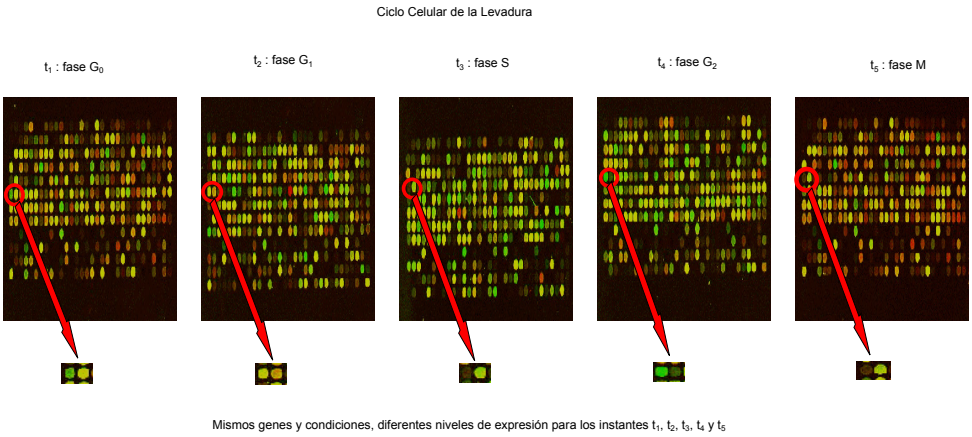



Figura 2.12: Experimento con microarray temporal

La correspondencia de los microarrays temporales con su equivalente digital se puede observar en Fig. 2.13. Un microarray temporal tiene $T$ puntos de tiempo, $G$ genes y $C$ condiciones, por lo tanto, para el análisis computacional, se agruparán varios experimentos de microarray (experimento de microarray temporal) para obtener varias tablas indexadas por gen (filas), condiciones experimentales (columnas) y puntos de tiempo (cada una de

las tablas). Cada una de las celdas de estas tablas es el nivel de expresión representado por un número real. Como se puede ver en el ejemplo de Fig. 2.13, la celda fila 2, columna 2, tiempo 1 es un número real que determina el nivel de expresión del gen número 2, bajo la condición experimental número 2 en el punto de tiempo 1.

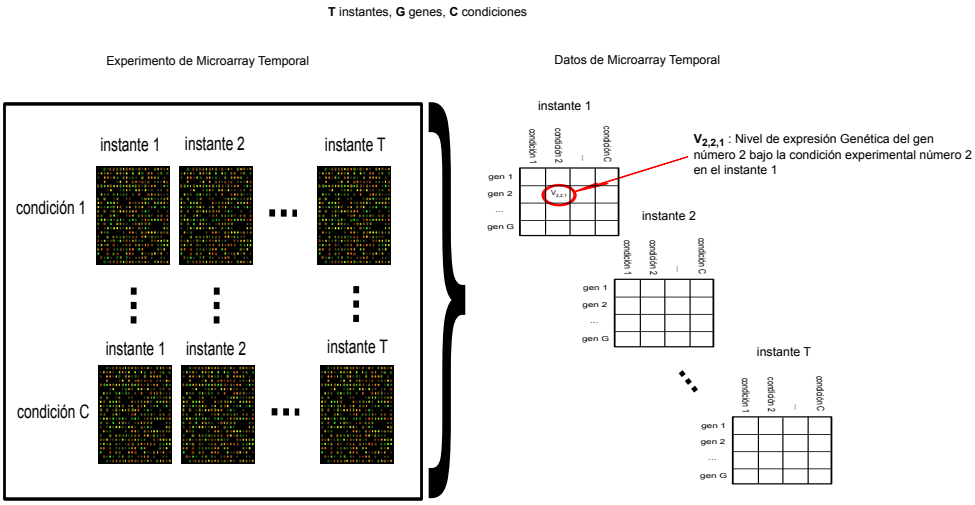



Figura 2.13: Datos de microarray temporal

## 2.4. Datos de magnitudes de seísmos

Estos conjuntos de datos constituyen un recurso formado por una agrupación de datos proveniente de estudios y mediciones de magnitudes de seísmos. Dichos datos han sido obtenidos del catálogo del *Centro Nacional de Información Geográfica* (en adelante CNIG) [55].

Desde el año 1961 el CNIG ha calculado la magnitud de los terremotos entre las coordenadas geográficas 5°N a 44°N y 10°W a 5°E produciendo catálogos de datos en una frecuencia semanal, mensual y global desde el año 1981 hasta la actualidad.

Para la estimación de las magnitudes, cinco tipos diferentes de correlaciones son tenidas en cuenta, estas son: Duración ($M_D(M - M_S)$) [50], Superficie de Onda ($m_{b,L_g}$(M-M$_S$)), Cuerpo de Onda ($m_b$(V-C)) [76], Superficie de Onda ($m_{b,L_g}$(L)) y Momento (M$_w$) [33]. Estas medidas de correlación son analizadas con profundidad en [53].

La estimación de estas magnitudes en el catálogo no son homogéneas ya que los seísmos de antes del año 1962 fueron calculados con procedimientos diferentes. Para garantizar la completitud del conjunto de datos sólo son incluidos los seísmos registrados después del año en el que el catálogo se compone. En ese año se incluyen todos los terremotos con magnitudes mayor o igual que la mínima magnitud registrada en dicho periodo.

El año 1978 es de completitud en el catálogo [54] con magnitud mínima de 3.0 por lo que, para este conjunto de datos, han sido tenidos en cuenta los seísmos desde ese año en adelante. El procedimiento usado para la localización de los epicentros es descrito en [49] que determina antiguos epicentros usando mapas isoseísmicos y, por último, se ha usado la aplicación *HYPO 71* basada en el tiempo de llegada de las ondas sísmicas a las estaciones y un modelo de corteza terrestre.

El siguiente paso, fue transformar el conjunto de datos 2D generado en 3D. En este proceso, varios atributos fueron generados para que las entradas puedan proporcionar información significativa. Para ello, primero se ordenaron todos los datos incluidos en el catálogo en 60×90 celdas representando cada una de ellas un área de 20×20 km$^2$ aproximadamente. Después, se enriqueció a cada celda con un conjunto de características que, de acuerdo con [61] y [68], son:

1. Número total de terremotos detectados, $N$.
2. Número de terremotos detectados con magnitud mayor o igual que 3.5 , $M_{3.5}$.
3. Número de terremotos detectados con magnitud mayor o igual que 4.0 , $M_{4.0}$.
4. Número de terremotos detectados con magnitud mayor o igual que 4.5 , $M_{4.5}$.
5. Número de terremotos detectados con magnitud mayor o igual que 5.0 , $M_{5.0}$.
6. Media de todos los epicentros de los terremotos, $\overline{D}$.
7. Máxima magnitud de terremotos, $M_{max}$.

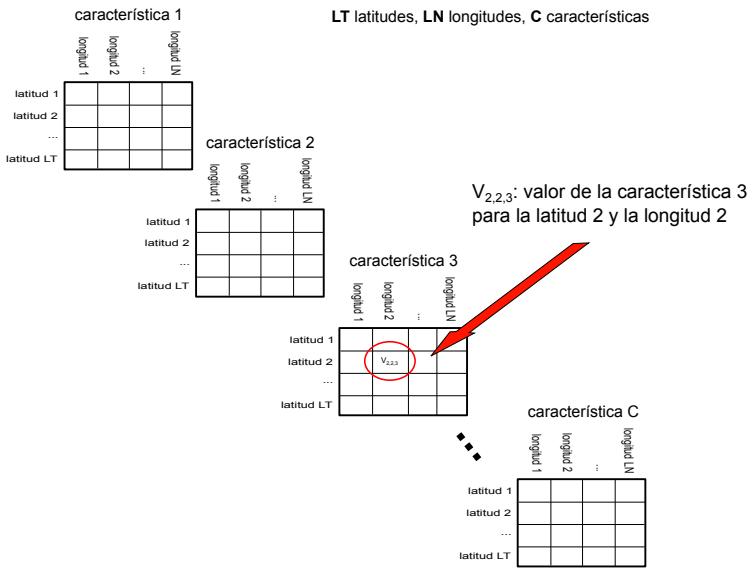



Figura 2.14: Conjunto de datos de magnitudes de seísmos

Por lo tanto, cada celda está definida como:

$$C_{i,j} = \{N^*, M^*_{3.5}, M^*_{4.0}, M^*_{4.5}, M^*_{5.0}, \overline{D^*}, M^*_{max}\} \tag{2.1}$$

donde $i \in [1, 90]$ y $j \in [1, 60]$ que indican la latitud y longitud relativa. La celda $(1, 1)$ corresponde a las coordenadas $(12°W, 33°N)$ mientras que $(90, 60)$ marca las coordenadas $(6°E, 45°N)$. En Fig. 2.14 vemos la disposición en tres dimensiones del conjunto de datos.

Vale la pena señalar que todas las características antes mencionadas se calcularon como sigue. Sea $C_{i,j}$ la celda objetivo, para calcular cada característica, tanto esta como las 24 celdas que la rodean se tienen en consideración, esto es, el conjunto de celdas incluido en el cuadrado con fronteras $C_{i-2,j-2}$ (esquina inferior izquierda) y $C_{i-2,j-2}$ (esquina superior derecha) son usadas para estimar la característica concreta de $C_{i,j}$.

## 2.5. Triclustering

De la necesidad de extender el Clustering al análisis de datos en tres dimensiones, nace el Triclustering. Esta técnica se define como un modelo para encontrar relaciones entre un subconjunto de dimensiones del total de los datos analizados. El Triclustering acepta una visión en función de la evolución de la técnica primigenia de la que procede, el cual es un punto de vista fundamental para entender su funcionamiento. Por ello, se centra la descripción de esta técnica en base a su desarrollo.

Los algoritmos de Clustering tradicionales tienen múltiples variantes y aplicaciones. Estos trabajan en todo el espacio de las dimensiones de los datos y, de manera general, puede considerarse como una técnica para agrupar objetos de cualquier naturaleza en base a sus características. Por ejemplo, podemos aplicar Clustering a un conjunto de clientes de unos grandes almacenes y agruparlos de forma que se parezcan entre sí.

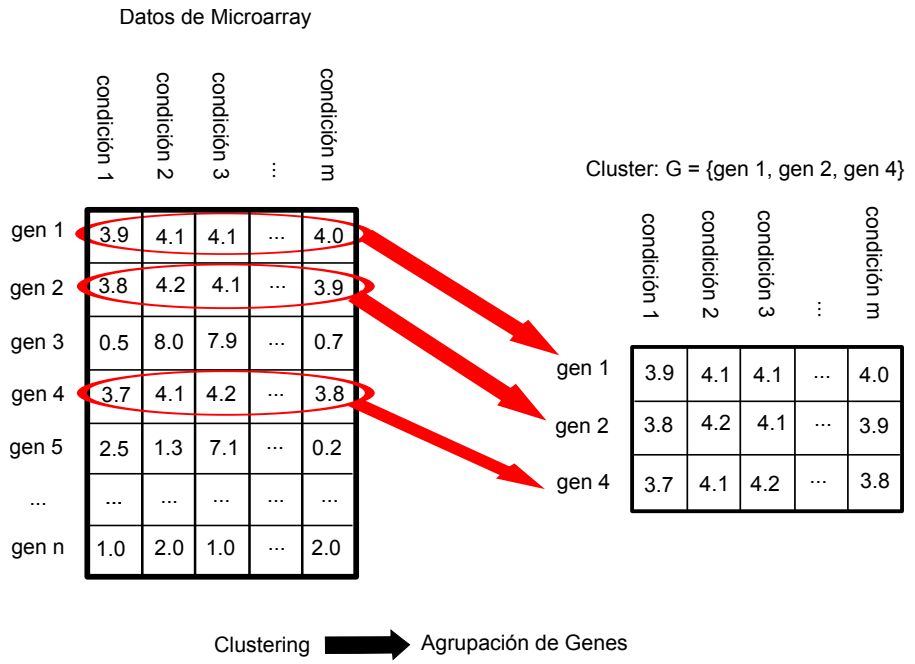



Figura 2.15: Clustering de datos de microarray

En el caso particular de los microarrays, disponemos de una base de datos en forma de tabla en la que cada celda representa el nivel de expresión genética de un gen concreto (fila) bajo una condición experimental concreta (columna), por tanto, al aplicar Clustering se obtienen agrupaciones de genes cuyo nivel de expresión genética es similar. Este hecho se observa en Fig. 2.15, en donde a partir de un microarray de entrada de $n$ genes y $m$ condiciones se agrupan los genes 1, 2 y 4 para formar el correspondiente cluster. Es destacable observar que los valores de expresión genética de los genes agrupados se pueden considerar similares (están en un rango entre 3.7 y 4.2) en comparación con el resto de niveles de expresión de los datos de entrada.

La aplicación del Clustering sobre un microarray da como resultado una serie de clusters que forman el modelo del mismo o, lo que es lo mismo, un conjunto de agrupaciones de genes que son similares entre sí en cuanto a su nivel de expresión genética.

El Biclustering va un paso más allá en la tarea de agrupar objetos. Con esta técnica no sólo agrupamos objetos a nivel de instancia como en el Clustering sino que también agrupamos a nivel de atributo. Siguiendo con el ejemplo de los grandes almacenes, al aplicar Biclustering, no sólo agrupamos clientes como concepto en general sino que además agrupamos por alguna de sus características, esto es, podemos agrupar clientes que tengan edades, gustos y salarios parecidos o clientes cuyo código postal y número de hijos sean parecidos.

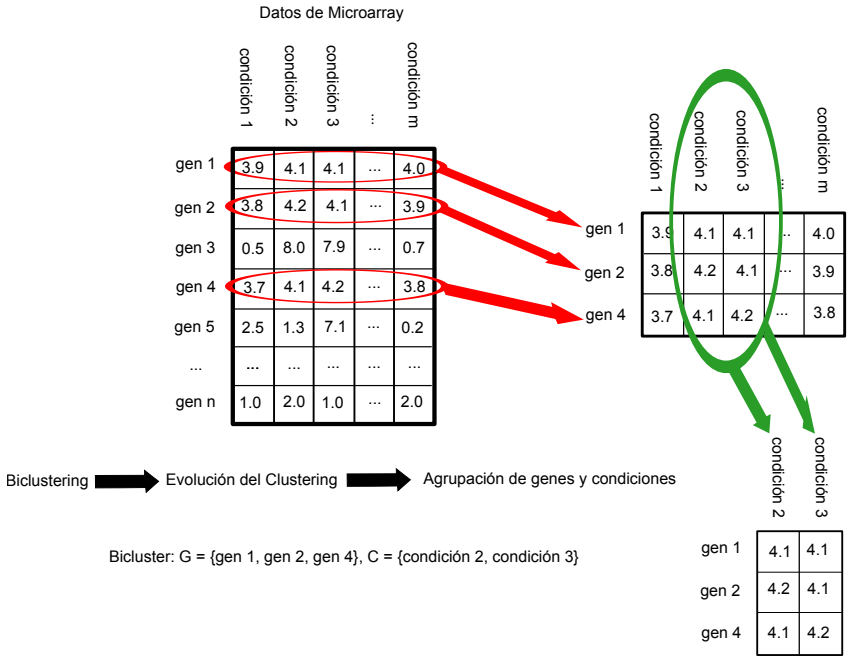



Figura 2.16: Biclustering de datos de microarray

Centrándonos de nuevo en el análisis de microarrays, al aplicar Biclustering, no sólo agrupamos genes con nivel de expresión parecido sino que también agrupamos condiciones experimentales. De esta forma, obtenemos biclusters que estarán formados por un subconjunto de genes y un subconjunto de condiciones experimentales de forma que los niveles de expresión que conforman estos subconjuntos son similares. Se puede decir que el Biclustering es de granularidad más fina que el Clustering en cuanto a profundidad dimensional en la búsqueda dentro del espacio de soluciones.

Como podemos observar en Fig. 2.16, partiendo de los datos de entrada de microarray y aplicando Biclustering, primero se observa una agrupación a nivel de gen para, seguidamente, realizar una agrupación a nivel de condición dejando patente el afinamiento dimensional anteriormente citado. Podemos ver que el resultado ya no es un grupo de genes como en el caso del Clustering, sino que es un grupo de genes y condiciones experimentales.

Vemos, igualmente, que los niveles de expresión genética del bicluster resultante conformado por los genes 1, 2 y 4 y las condiciones 2 y 3 son muy similares entre sí (están en un rango entre 4.1 y 4.2), lo que acentúa la característica de precisión dimensional de esta técnica.

La siguiente evolución en la técnica es conocida como el Triclustering. Esta técnica nace por necesidad de análisis de datos cuyos valores varían en el tiempo por lo que añaden una nueva dimensión. Debido a este hecho, las técnicas de Clustering y Biclustering no proporcionan un análisis completo de estos datos.

Al aplicar Triclustering a conjuntos de datos variantes en el tiempo, estamos agrupando no sólo a nivel de instancia y característica, sino que también a nivel de instante de tiempo en el que dichos datos fueron adquiridos. Continuando con el ejemplo de los grandes almacenes, no sólo agrupamos clientes según su concepto en general y según alguna de sus características sino que también agruparemos por instante de tiempo en el que los datos fueron adquiridos o medidos, de esta forma, podemos agrupar clientes no sólo que tengan gustos, salarios, código postal y número de hijos

parecidos sino que también que dichas características estén comprendidas en un periodo de tiempo concreto por ejemplo entre abril y mayo.

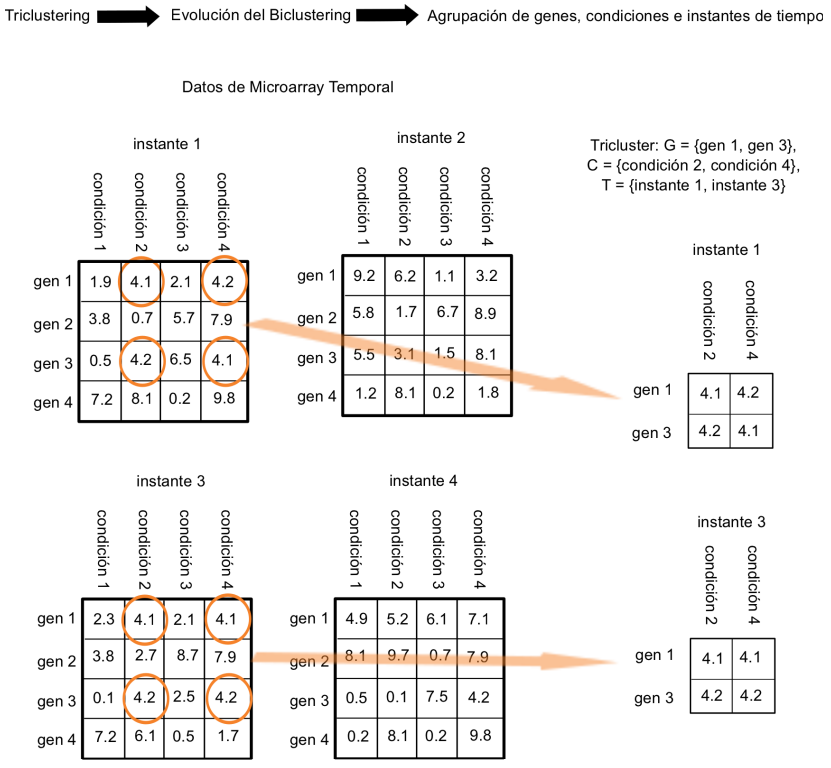



Figura 2.17: Triclustering de datos de microarray

En el caso particular en el que los datos de entrada son los microarrays temporales, y a diferencia de los dos anteriores, disponemos de una base de datos en forma de cubo en la que cada celda representa no sólo el nivel de expresión genética de un gen concreto (fila) bajo una condición experimental concreta (columna) sino que también en un instante de tiempo concreto (profundidad).

Por tanto, al aplicar Triclustering estamos obteniendo agrupaciones de genes bajo condiciones experimentales concretas y en instantes de tiempo concretos cuyo nivel de expresión genética es similar.

Todos estos conceptos quedan patentes en Fig. 2.17 en la que se observa como la entrada al análisis son datos de microarray temporal que incluye una nueva dimensión de tiempo diferenciándose con los dos casos anteriores ya que forma un "cubo" de datos en lugar de una "tabla". Como resultado se obtiene un subconjunto de genes (el 1 y el 3), un subconjunto de condiciones (la 2 y la 4) y un subconjunto de instantes de tiempo (el 1 y el 3) cuyos niveles de expresión genética que lo conforman son de valores muy similares entre sí (están en un rango entre 4.1 y 4.2).

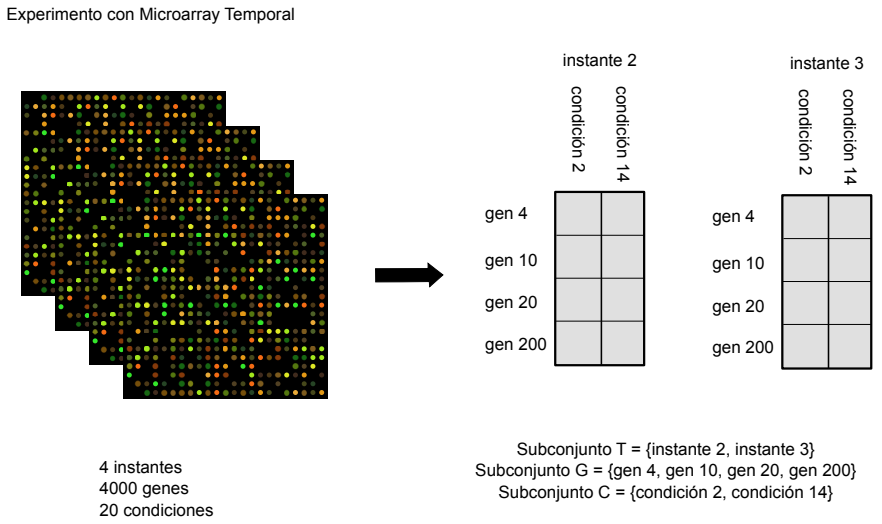



Figura 2.18: Tricluster

El resultado de la aplicación de Triclustering sobre microarrays temporales es un conjunto de triclusters. Como podemos ver en Fig. 2.18, de un experimento con microarray temporal se obtiene un modelo compuesto por una serie de triclusters que están formados por un subconjunto de genes, condiciones experimentales e instantes de tiempo. El objetivo que se persi-

gue en la línea de investigación que presentamos es que los triclusters del modelo resultante proporcionen un patrón de comportamiento de los niveles de expresión de los genes que agrupan, bajo condiciones experimentales concretas y en puntos de tiempo concretos.

El patrón de comportamiento genético definido como agrupación de genes es el objeto de búsqueda cuando hablamos de encontrar información oculta, válida y útil. La idea es agrupar los genes de un microarray temporal que se comporten de manera similar en unas condiciones experimentales concretas y en unos puntos de tiempo concretos. Los patrones de comportamiento genético se representan gráficamente mostrando tres disposiciones de los valores que lo componen.

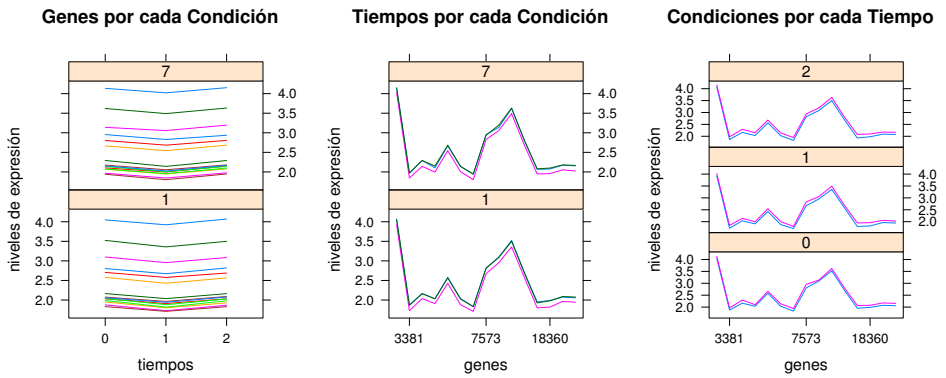



Figura 2.19: Tricluster como patrón de comportamiento genético

En la primera, para cada condición experimental se representan los tiempos en el eje X, los niveles de expresión en el eje Y y cada línea de la gráfica representa un gen, en la segunda, para cada condición experimental se representan los genes en el eje X, los niveles de expresión en el eje Y y cada línea de la gráfica representa un punto de tiempo y en la tercera, para cada punto de tiempo se representan los genes en el eje X, los niveles de expresión en el eje Y y cada línea de la gráfica representa una condición.

En Fig. 2.19 se puede observar un ejemplo de tricluster en forma de patrón de comportamiento extraído de un experimento de microarray que estudia la degeneración celular de las retinas en ratones (*Mus musculus*) [20] de 22690 genes, 8 condiciones experimentales y 4 puntos de tiempo.

## 2.6. Algoritmos Genéticos

Los algoritmos genéticos se consideran técnicas computacionales bioinspiradas que aplican una metaheurística o procedimiento inteligente basado en el conocimiento experto sobre el dominio del problema que se quiere atacar. Dicho conocimiento se basa en la *Teoría de la Evolución* de Charles Darwin [18] y se fundamenta en hacer evolucionar las potenciales soluciones o población de individuos hacia la solución óptima aplicando elementos de combinación, elitismo y aleatoriedad a las mismas.

Fueron inicialmente presentados por Holland en 1975 [66] donde estableció los conceptos teóricos y catalogó las potenciales aplicaciones. Más tarde Goldberg [23] profundizó en los múltiples campos de aplicación de los algoritmos genéticos. Otras publicaciones más orientadas al mundo académico y altamente ilustrativas pueden consultarse en [52, 60].

Los algoritmos genéticos, de manera general, siguen el patrón mostrado en Fig. 2.20. En ella podemos observar la representación genética y el propio proceso evolutivo como partes fundamentales de la técnica.

La organización genética más común establece el *gen* como unidad mínima de información, el *individuo* como conjunto de genes y la *población* como conjunto de individuos. Los distintos operadores evolutivos se encargan de hacer evolucionar la información contenida en los genes de los individuos de la población intercambiando dicho "material genético". De manera general, el proceso evolutivo consiste en la aplicación de todos estos operadores durante un número concreto de generaciones conformando un proceso iterativo hasta obtener la solución.

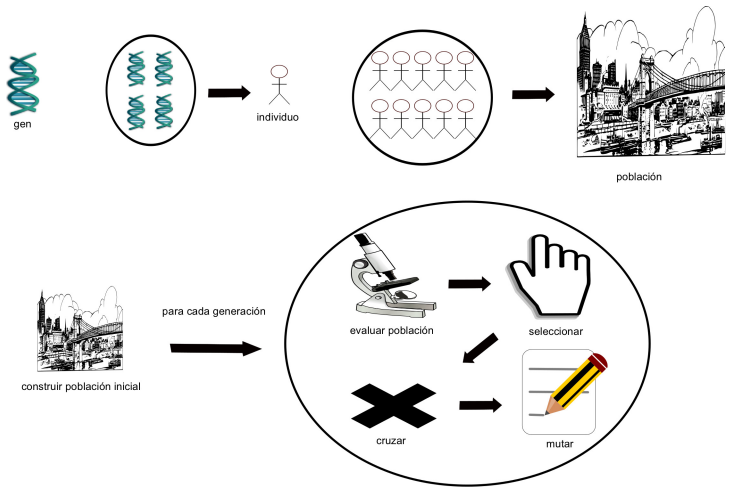



Figura 2.20: Algoritmo genético

Los operadores genéticos se definen como sigue:

- Construcción de la población inicial: dicha población contendrá individuos que, a su vez, estarán compuestos de genes.
- Evaluación de la población: cada individuo de la población será evaluado para medir la bondad del mismo como solución al problema.
- Selección: Elegir los individuos adecuados para pasar a la siguiente generación.
- Cruce: Combinar los genes de cada individuo se combinan para formar otros nuevos.
- Mutación: Cambiar puntualmente el conjunto de los genes de algunos individuos asegurando variabilidad en la siguiente generación.

Los campos así como las posibilidades de aplicación del paradigma de algoritmo genético son innumerables gracias al alto grado de funcionalidad

de dicho esquema y gracias a su probada eficiencia en la búsqueda de soluciones. Dicha eficiencia es debida a la capacidad del operador de cruce para la explotación de soluciones y la del operador de mutación para la exploración de las mismas.

Los problemas que precisan de soluciones aproximadas debido a la imposibilidad de encontrar una solución óptima total son susceptibles de ser atacados por dicha metaheurística. Así pues, esta técnica es utilizada en campos de aplicación tan dispares como la ingeniería eléctrica, estudios de mercado, bioinformática, aprendizaje automático, etc.

## 2.7. Gene Ontology

*The Gene Ontology Project* (en adelante *GO)* [16] supone un esfuerzo colaborativo para la construcción y el uso de ontologías con el fin de facilitar la significativa tarea biológica de anotación de genes y sus productos para una amplia variedad de organismos. Las entidades participantes en dicho proyecto incluyen grandes bases de datos de modelos de organismos además de centros de recursos bioinformáticos [17].

*GO* proporciona un lenguaje sistemático u ontología para la descripción de atributos de genes y sus productos divididos en tres dominios clave compartidos por todos los organismos: funciones moleculares, procesos biológicos y componentes celulares.

Las anotaciones de gen-término registradas en *GO* suponen una herramienta muy útil para obtener el significado funcional y biológico de grandes conjuntos de datos como los microarrays. Del mismo modo, también facilita la organización de los datos provenientes de nuevos y completamente anotados genomas y la comparación de la información biológica.

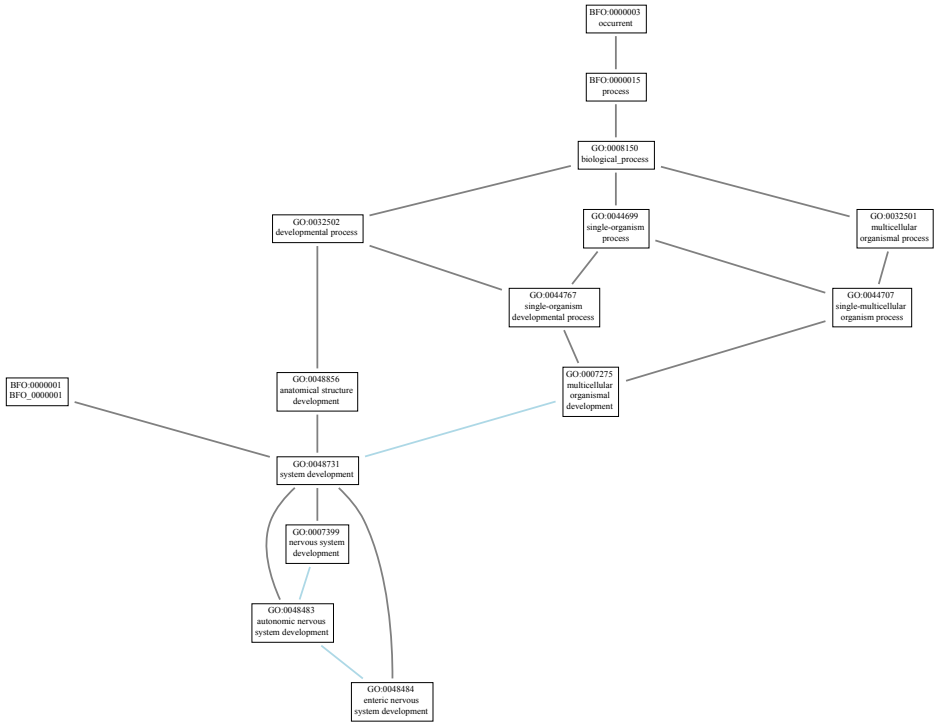



Figura 2.21: Árbol de términos *GO*

Los términos *GO* están organizados en redes jerárquicas donde cada nivel se corresponde con una diferente especificación de los términos, con lo cual, los términos de niveles más altos son más generales que los niveles más bajos (Fig. 2.21).

Desde el punto de vista computacional estas jerarquías están estructuradas como árboles (grafos conexos sin ciclos dirigidos). Existen numerosos enfoques en la literatura [5, 62, 51] cuyo objetivo es el de producir el análisis estadístico para la sobre-representación de los términos *GO* en conjuntos de genes o proteínas derivados de un experimento y obtener la validación biológica del mismo.

## 2.8. Coeficientes de Correlación

El término correlación se define como una relación estadística que implica dependencia. Ejemplos de fenómenos dependientes incluyen la correlación entre las estructuras físicas de los padres y su descendencia o la correlación entre la demanda de un producto y su precio.

La correlación es útil ya que puede indicar una relación predictiva que puede ser explotada en la práctica. Por ejemplo, una planta eléctrica puede producir menos potencia a mitad del día basándose en la correlación entre demanda eléctrica y el clima.

Formalmente, se dice de Correlación para cualquier situación en la que variables aleatorias no satisfagan la condición matemática de independencia probabilista. Puede referirse a cualquier exclusión de independencia de dos o más variables aleatorias, pero técnicamente se refiere a varios de los diversos tipos de relación entre valores medios más especializados.

Existen varios coeficientes de correlación que miden el grado de dependencia de dos variables aleatorias. Dos de los más utilizados son:

- Coeficiente de correlación de Pearson [57]: mide la dependencia lineal de dos variables proporcionando un valor en el rango $[-1, 1]$ donde $-1$ indica correlación totalmente negativa, 0 indica no correlación y 1 correlación totalmente positiva.
- Coeficiente de correlación de Spearman [70]: mide el grado de bondad de la relación de dos variables aleatorias que pueden ser descritas usando una función monótona proporcionando un valor en el rango $[-1, 1]$ donde $-1$ y 1 indican que una de las variables tiene una relación monótona y perfecta con respecto a la otra.

# Capítulo 3

# Resumen de las publicaciones

En este capítulo se proporcionará una visión global y resumida del presente trabajo de investigación. En primera instancia se pondrá el foco en cada uno de los hitos conseguidos en este trabajo en orden de desarrollo. Teniendo como punto de partida el análisis del algoritmo *TriGen* (Sección 3.1), se prosigue dando las claves fundamentales de los tres desarrollos posteriores: la medidas $MSR_{3D}$ (Sección 3.2), $LSL$ (Sección 3.4) y $MSL$ (Sección 3.5); para las dos últimas medidas se expondrá previamente el concepto de *Vistas Gráficas* de un tricluster (Sección 3.3). Seguidamente se analizará el desarrollo de medida global de calidad de un tricluster $TRIQ$ (Sección 3.6) y la aplicación de este marco de trabajo al problema de la zonificación de seísmos (Sección 3.7). Para finalizar el capítulo se expondrá una discusión conjunta de los resultados obtenidos en los distintos desarrollos de esta investigación (Sección 3.8).

## 3.1. Algoritmo *TriGen*

El primer desarrollo de este trabajo de investigación, el algoritmo *TriGen*, fue inicialmente presentado en [31, 29, 30] (Sección 5) para, posteriormente, ser publicado su diseño, implementación y efectividad en [28] (Sección 4.1). En las posteriores publicaciones, aunque no centradas en el diseño del algoritmo, se describen las modificaciones del mismo a nivel funcional y metodológico.

*TriGen* ("Triclustering Genético") hace uso del paradigma bio-inspirado de la metaheurística evolutiva de los algoritmos genéticos (Sección 2.6) para extraer triclusters de experimentos de microarrays adquiridos en distintos instantes de tiempo (Sección 2.3). Los triclusters resultantes conforman un grupo de patrones de comportamiento de los genes en el espacio tridimensional, esto es, teniendo en cuenta, además, las condiciones experimentales y los puntos temporales.

Como se puede observar en Fig. 3.1 el proceso evolutivo del algoritmo *TriGen* se compone de varios operadores: un paso de generación de la población inicial en el que se creará el grupo de individuos o triclusters que posteriormente serán evolucionados; un paso de evaluación que constituye el núcleo del algoritmo y en el que se evalúa la bondad de cada tricluster de cada generación; un método de cruce para crear las conexiones necesarias entre cada par de individuos para mezclar el material genético y un método de mutación que produce cambios puntuales en los individuos para asegurar la variabilidad genética de las próximas generaciones.

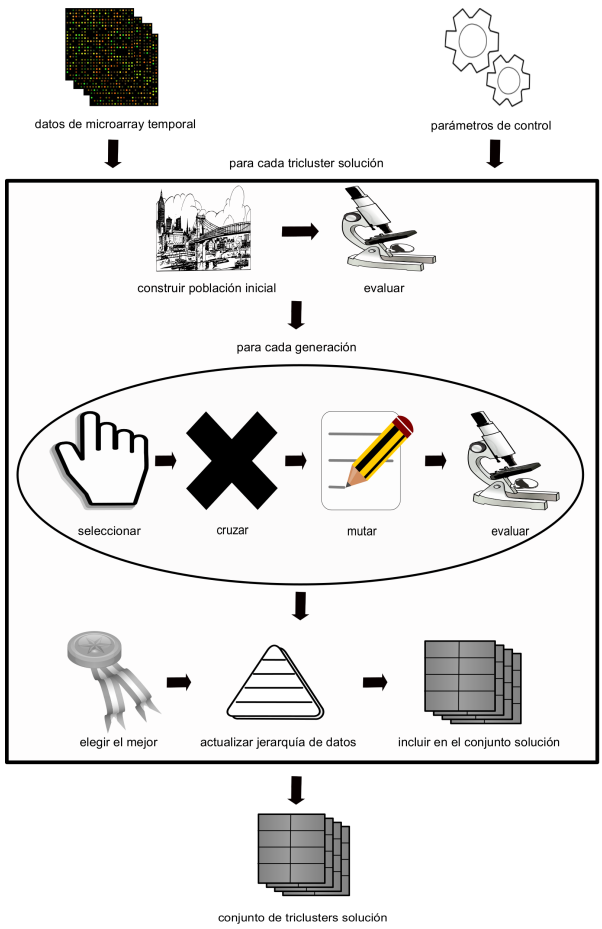



Figura 3.1: Algoritmo *TriGen*

Este proceso evolutivo tiene un elemento de apoyo, la jerarquía de datos, cuyo objetivo es evitar el solapamiento de las soluciones retro-alimentando la construcción de cada población con el conocimiento aportado por la soluciones ya encontradas.

### 3.1.1. Entrada y Salida

El algoritmo *TriGen* tiene dos argumentos de entrada:

- $D$: El dataset de entrada que contiene los valores de expresión de los genes sdel experimento temporal con microarrays. El dataset tendrá un número $D_G$ de genes, $D_C$ de condiciones experimentales y $D_T$ de puntos de tiempo formando un cubo de datos. Por lo tanto, cada celda $[i, j, k]$ donde $i \in D_G$, $j \in D_C$ y $k \in D_T$ contiene el valor del nivel de expresión del gen $i$ bajo la condición experimental $j$ en el instante $k$.
- $P$: Conjunto de parámetros que controlan la ejecución del algoritmo (tabla 3.1). Éstos controlan el número de soluciones o triclusters que el algoritmo debe encontrar ($N$), el número de procesos evolutivos a ejecutar para cada solución ($G$), el número de individuos que contendrá la población a evolucionar ($I$) y el factor de aleatoriedad que se incluirá en la generación de la misma ($Ale$). Del mismo modo, controlan la ejecución de los operadores genéticos con el ratio de selección ($Sel$) y la probabilidad que tiene un individuo de ser mutado ($Mut$). También son controlados el efecto del la función de fitness ($w_f$), el tamaño de los triclusters ($w_g$, $w_c$, $w_t$) y el solapamiento de los mismos ($wo_g$, $wo_c$, $wo_t$) en el paso de evaluación de la población.

La salida del algoritmo será un conjunto de $N$ triclusters, $SOL = \{TRI_1, TRI_2, \ldots, TRI_N\}$ en el que cada $TRI_i \, \epsilon \, SOL$ está compuesto por un subconjunto de genes $TRI_G$, condiciones experimentales $TRI_C$ y puntos de tiempo $TRI_T$ del dataset de entrada $D$ que tendrá la mejor puntuación en su población resultante del paso de evaluación.

Tabla 3.1: Parámetros de control del algoritmo *TriGen*

| Parámetro | Descripción |
|---|---|
| $N$ | Número de triclusters a extraer |
| $G$ | Número de generaciones |
| $I$ | Número de individuos en la población |
| $Ale$ | Factor de aleatoriedad |
| $Sel$ | Ratio de selección |
| $Mut$ | Probabilidad de mutación |
| $w_f$ | Peso de la función fitness |
| $w_g$ | Peso para el número de genes |
| $w_c$ | Peso para el número de condiciones |
| $w_t$ | Peso para el número de tiempos |
| $wo_g$ | Peso para el solapamiento de genes |
| $wo_c$ | Peso para el solapamiento de condiciones |
| $wo_t$ | Peso para el solapamiento de tiempos |

### 3.1.2. Codificación de los Individuos y Operadores Genéticos

Cada individuo de la población representa a un tricluster $TRI$ el cual es una solución potencial. Éste contiene el material genético que será manipulado por los operadores genéticos. Dicho material genético está compuesto por tres estructuras secuenciales: una estructura de genes $TRI_G$, otra de condiciones experimentales $TRI_C$ y otra de puntos de tiempo $TRI_T$; todas ellas subconjuntos del dataset de entrada $D$ y constituidas como sigue:

El subconjunto $TRI_G$ lo componen una secuencia, ordenada según su posición en el dataset, de genes provenientes del conjunto global de genes $D_G$ del dataset de entrada (Ec. 3.1).

$$\begin{gathered} TRI_G =< g_{i_1}, g_{i_2}, \ldots, g_{i_{|TRI_G|}} >, \\ \forall g_{i_j} \in TRI_G \; (g_{i_j} < g_{i_{j+1}}) \; \wedge (g_{i_j} \in D_G = \{g_{i_1}, g_{i_2}, \ldots, g_{i_{|D_G|}}\}) \end{gathered} \tag{3.1}$$

El subconjunto $TRI_C$ lo componen una secuencia, ordenada según su posición en el dataset, de condiciones provenientes del conjunto global de condiciones $D_C$ del dataset de entrada (Ec. 3.2).

$$\begin{gathered} TRI_C =< c_{i_1}, c_{i_2}, \ldots, c_{i_{|TRI_C|}} >, \\ \forall c_{i_j} \in TRI_C \; (c_{i_j} < c_{i_{j+1}}) \; \wedge (c_{i_j} \in D_C = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{|D_C|}}\}) \end{gathered} \tag{3.2}$$

El subconjunto $TRI_T$ lo componen una secuencia, ordenada según su posición en el dataset, de puntos temporales provenientes del conjunto global de puntos temporales $D_T$ del dataset de entrada (Ec. 3.3).

$$\begin{gathered} TRI_T =< t_{i_1}, t_{i_2}, \ldots, t_{i_{|TRI_T|}} >, \\ \forall t_{i_j} \in TRI_T \; (t_{i_j} < t_{i_{j+1}}) \; \wedge (t_{i_j} \in D_T = \{t_{i_1}, t_{i_2}, \ldots, t_{i_{|D_T|}}\}) \end{gathered} \tag{3.3}$$

Cada población del algoritmo se compone de varios individuos organizados como se representa en Fig. 3.2.

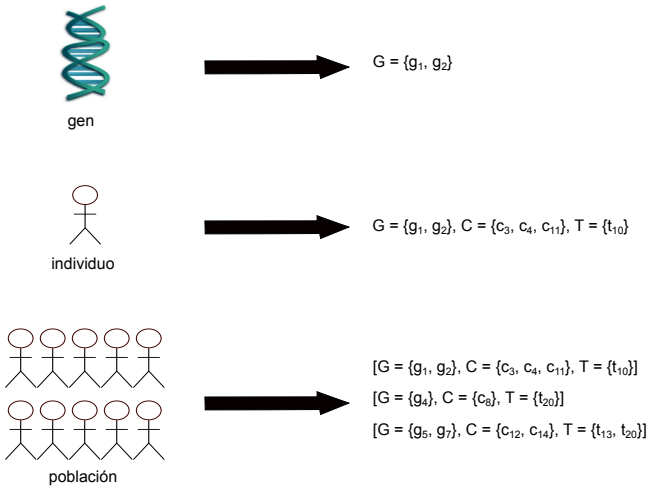



Figura 3.2: Codificación de los individuos

El sistema de control de solapamiento del algoritmo *TriGen* es llamado *Jerarquía de Datos* y consiste en mantener el número de ocurrencias de genes, condiciones experimentales y puntos de tiempo del dataset $D$ para cada solución encontrada de manera jerárquica situando las menos visitadas en el nivel más alto de la jerarquía. El operador de población inicial usará dicha estructura en la creación de los individuos con mínimo solapamiento. La *Jerarquía de Datos* es actualizada cada vez que una solución es encontrada como resultado del proceso evolutivo.

Los operadores genéticos del algoritmo *TriGen* son los siguientes:

**Población inicial** Con este método $I$ individuos son generados teniendo en cuenta el factor de aleatoriedad *Ale*. Un tanto por ciento *Ale* de individuos son creados de manera aleatoria mediante dos métodos: la mitad del porcentaje son generados puramente de manera aleatoria, esto es, un subconjunto aleatorio de genes $TRI_G$, condiciones $TRI_C$ y tiempos $TRI_T$ son elegidos de $D$; la otra mitad es creada también de manera aleatoria pero controlando que los genes $TRI_G$, condiciones $TRI_C$ y tiempos $TRI_T$ sean contiguos de tal forma que formen un cubo compacto. El resto de los individuos son creados teniendo en cuenta las soluciones anteriormente descubiertas para controlar el solapamiento de los mismos acorde a la estructura *Jerarquía de Datos* anteriormente descrita.

**Evaluación** Este operador evalúa cada individuo en base a la función de evaluación $FF(TRI)$ a optimizar. Como se puede observar en la Ec. 3.4, $FF(TRI)$ está compuesta de siete factores.

$$\begin{aligned} FF(TRI) = \frac{1}{w_f + w_g + w_c + w_t + wo_g + wo_c + w_t} * \\ \{w_f * FITNESS(TRI) + \\ w_g * \left(1 - \frac{|TRI_G|}{|D_G|}\right) + w_c * \left(1 - \frac{|TRI_C|}{|D_C|}\right) + w_t * \left(1 - \frac{|TRI_T|}{|D_T|}\right) + \\ wo_g * \frac{R_G(TRI, SOL)}{|TRI_G| * |SOL|} + wo_c * \frac{R_C(TRI, SOL)}{|TRI_C| * |SOL|} + wo_t * \frac{R_T(TRI, SOL)}{|TRI_T| * |SOL|}\} \end{aligned} \tag{3.4}$$

El principal es $FITNESS(TRI)$ que proporciona el valor normalizado de la medida de evaluación de $TRI$ que, para este trabajo de investigación, se han desarrollado tres: $MSR_{3D}$, $LSL$ y $MSL$ que son descritas en el presente documento en las secciones 3.2, 3.4 y 3.5 respectivamente. Cuando $w_f$ es incrementado, se favorece la búsqueda de triclusters con mayor $FITNESS(TRI)$.

El siguiente grupo de tres factores $1 - \frac{|TRI_G|}{|D_G|}$, $1 - \frac{|TRI_C|}{|D_C|}$ y $1 - \frac{|TRI_T|}{|D_T|}$ corresponden al control del tamaño del tricluster y miden el número de genes condiciones y tiempos de $TRI$ ($|TRI_{G,C,T}|$) relativos al tamaño del dataset de entrada ($|D_{G,C,T}|$). Al ser $FITNESS(TRI)$ una función a minimizar se resta 1 menos cada proporción de tamaño para favorecer mayores tamaños con el incremento de los pesos $w_g$, $w_c$ o $w_t$.

Los otros tres factores $\frac{R_G(TRI,SOL)}{|TRI_G|*|SOL|}$, $\frac{R_C(TRI,SOL)}{|TRI_C|*|SOL|}$ y $\frac{R_T(TRI,SOL)}{|TRI_T|*|SOL|}$ miden las repeticiones de genes, condiciones y puntos de tiempo de $TRI$ en el conjunto de soluciones ya encontradas $SOL$ ($R_{G,C,T}(TRI, SOL)$) proporcionalmente calculadas teniendo en cuenta todos los genes, condiciones o tiempos repetidos en $SOL$ ($|TRI_{G,C,T}|*|SOL|$) con el objetivo de favorecer el menor solapamiento de $TRI$ con el incremento de los pesos $wo_g$, $wo_c$ o $wo_t$.

Una configuración por defecto para $w_f$, $w_g$, $w_c$, $w_t$, $wo_g$, $wo_c$ y $wo_t$ consiste en fijar $w_f$ a 0.8 y distribuir 0.2 entre $w_g$, $w_c$, $w_t$, $wo_g$, $wo_c$ y $wo_t$ de tal forma que se favorece la búsqueda de soluciones en las que predomine el valor de la función $FITNESS(TRI)$ y el resto del peso queda repartido proporcionalmente entre los miembros que controlan el tamaño y el solapamiento de dichas soluciones.

**Selección** Tres grupos de individuos son seleccionados de manera aleatoria y ordenados de menor a mayor acorde a la función de evaluación, después, se realiza una selección de individuos aleatoria de entre estos tres grupos. El ratio de selección $Sel$ indica la cantidad de individuos que serán seleccionados y que pasarán a la siguiente generación mientras que el resto ($I$ - $\#Individuos\ seleccionados$), que completará la población promocionada, serán creados en base al operador de cruce.

**Cruce** Para completar la siguiente generación, se crearan nuevos individuos con este operador de la siguiente forma: dos individuos, los padres $A$ y $B$, son combinados para crear dos nuevos descendientes hijo 1 e hijo 2. Los padres son elegidos de manera aleatoria y el material genético es combinado mediante el método unipuntual para los genes $TRI_G$, condiciones $TRI_C$ y tiempos $TRI_T$ mezclando los elementos de ambos hijos.

**Mutación** Un individuo puede ser mutado según la probabilidad que marca el parámetro $Mut$. Para cada individuo de la población, la probabilidad será verificada y, en caso de ser satisfecha, una de nueve posibles acciones es realizada. Estas acciones son: añadir de manera aleatoria un nuevo gen a $TRI_G$, una nueva condición a $TRI_C$ o un nuevo punto de tiempo a $TRI_T$, eliminar de manera aleatoria un gen de $TRI_G$, una condición de $TRI_C$ o un punto de tiempo de $TRI_T$ o cambiar de manera aleatoria un gen de $TRI_G$, una condición de $TRI_C$ o un punto de tiempo de $TRI_T$. La elección de estas acciones es también aleatoria. Para el caso de la adición de un nuevo gen, condición o tiempo, el operador comprueba si el nuevo elemento está o no en el individuo.

## 3.2. Mean Square Residue 3D ($MSR_{3D}$)

El residuo cuadrático medio (mean squared residue en inglés, $MSR$) fue introducido por Cheng y Church en [13]. Esta medida fue propuesta para evaluar la calidad de los biclusters extraídos de datos de expresión genética basados en la homogeneidad de los mismos. La definición formal de esta medida puede observarse en la Ec. 3.5:

$$MSR(BC) = \frac{\sum_{g \epsilon G, c \epsilon C} r_{gc}^2}{\#G * \#C} \tag{3.5}$$

donde $r_{gc}$ es definido como:

$$r_{gc} = BC_v(g,c) - M_G(c) - M_C(g) - M_{GC} \tag{3.6}$$

Cada uno de los términos de Ec. 3.5 y Ec. 3.6 son definidos como sigue:

- $BC$: Bicluster a evaluar.
- $G$: Subconjunto de genes de $BC$.
- $C$: Subconjunto de condiciones de $BC$.
- $\#G$: Número de genes en $BC$.
- $\#C$: Número de condiciones en $BC$.
- $BC_v(g,c)$: Nivel de expresión del gen $g$ bajo la condición $c$ en $BC$.
- $M_G(c)$: Media de los valores de la condición $c$ para todos los genes en $BC$.
- $M_C(g)$: Media de los valores del gen $g$ bajo todas las condiciones en $BC$.
- $M_{GC}$: Media de todos los valores en $BC$.

Se puede decir que $MSR$ mide la homogeneidad para un bicluster dado, basándose en la diferencia de cada valor de expresión genética de cada gen $BC_{v(i,j)}$ con la media de los valores de los genes $M_{G(j)}$, condiciones $M_{C(i)}$ y genes y condiciones $M_{GC}$.

Los biclusters con valores próximos de $MSR$ a cero tendrán mayor homogeneidad.

El análisis de $MSR$ y su adaptación a las tres dimensiones para poder ser utilizada para medir la homogeneidad de los triclusters ha sido llevado a cabo en esta investigación en [26] (Sección 4.2). En nuestra propuesta, definimos formalmente $MSR_{3D}$ como se muestra en la Ec. 3.7:

$$MSR_{3D}(TC) = \frac{\sum_{g\epsilon G, c\epsilon C, t\epsilon T} r_{gct}^2}{\#G * \#C * \#T} \tag{3.7}$$

donde $r_{gct}$ es definido como:

$$\begin{aligned} r_{gct} = TC_v(g,c,t) + M_{CT}(g) + M_{GT}(c) + M_{GC}(t) - \\ M_G(c,t) - M_C(g,t) - M_T(g,c) - M_{GCT} \end{aligned} \tag{3.8}$$

Cada miembro de las ecuaciones 3.7 y 3.8 son definidos como:

- $TC$: Tricluster a evaluar.
- $G$: Subconjunto de genes de $TC$.
- $C$: Subconjunto de condiciones de $TC$.
- $T$: Subconjunto de tiempos de $TC$.
- $\#G$: Número de genes en $TC$.
- $\#C$: Número de condiciones en $TC$.
- $\#T$: Número de tiempos en $TC$.

- $TC_v(g, c, t)$: Nivel de expresión del gen $g$ bajo la condición $c$ en el tiempo $t$ para $TC$.
- $M_{CT}(g)$: Media de todas las condiciones, teniendo en cuenta todos los tiempos, para el gen $g$ en $TC$.
- $M_{GT}(c)$: Media de todos los genes, teniendo en cuenta todos los tiempos, para la condición $c$ en $TC$.
- $M_{GC}(t)$: Media de todos los genes, bajo todas las condiciones en el tiempo $t$ en $TC$.
- $M_G(c, t)$: Media de los valores de la condición $c$ y el tiempo $t$ bajo todos los genes en $TC$.
- $M_C(g, t)$: Media de los valores del gen $g$ y el tiempo $t$ bajo todas las condiciones en $TC$.
- $M_T(g, c)$: Media de los valores del gen $g$ y la condición $c$ bajo todas los tiempos en $TC$.
- $M_{GCT}$: Media de todos los valores de $TC$.

$MSR_{3D}$ mide la homogeneidad de un tricluster dado basado en la diferencia del nivel de expresión genética individual, $TC_v(i, j, k)$, la media de todas las condiciones en todos los tiempos para un gen $g$, $M_{CT}(g)$, la media de todos los genes en todos los tiempos para una condición $c$, $M_{GT}(c)$, y la media de todos los genes bajo todas las condiciones para un tiempo $t$, $M_{GC}(t)$, con la media de una condición $c$ y un tiempo $t$ bajo todos los genes, $M_G(c, t)$, la media de un gen $g$ y un tiempo $t$ bajo todas las condiciones, $M_C(g, t)$, la media de un gen $g$ y una condición $c$ bajo todos los tiempos, $M_T(g, c)$, y la media de todos los valores en $TC$, $M_{GCT}$.

Los triclusters con valores de $MSR_{3D}$ próximos a cero tendrán mayor homogeneidad. Debido a su formulación $MSR_{3D}$ es capaz de encontrar genes negativamente correlacionados.

## 3.3. Vistas Gráficas de un Tricluster

Con el objetivo de explicar las dos próximas propuestas, $LSL$ (Sección 3.4) y $MSL$ (Sección 3.5), definimos las vistas gráficas de un tricluster $TRI$ como $TRI_{xop}$ [25].

Para una vista gráfica $TRI_{xop}$, $x$ representa las coordenadas en el eje $X$ y $o$ las líneas gráficas representadas en tantos paneles como indique $p$ como puede observarse en Fig. 3.3. Cada uno de los elementos $x$, $o$ y $p$ serán, indistintamente, los genes $G$, las condiciones experimentales $C$ o los puntos de tiempo $T$ de $TRI$. Para analizar visualmente el patrón de comportamiento descrito por un tricluster, tenemos en cuenta tres vistas gráficas:

- $TRI_{gct}$ $(x = G, o = C, p = T)$: un panel por cada tiempo, los genes en el eje X, los niveles de expresión en el eje Y y las condiciones como líneas gráficas representadas.
- $TRI_{gtc}$ $(x = G, o = T, p = C)$: un panel por cada condición, los genes en el eje X, los niveles de expresión en el eje Y y los tiempos como líneas gráficas representadas.
- $TRI_{tgc}$ $(x = T, o = G, p = C)$: un panel por cada condición, los tiempos en el eje X, los niveles de expresión en el eje Y y los genes como líneas gráficas representadas.

Con $TRI_{gct}$ y $TRI_{gtc}$ podemos analizar como cada nivel de expresión varía a través de las condiciones y tiempos respectivamente. $TRI_{tgc}$ representa como cada gen varía a través del tiempo para cada condición.

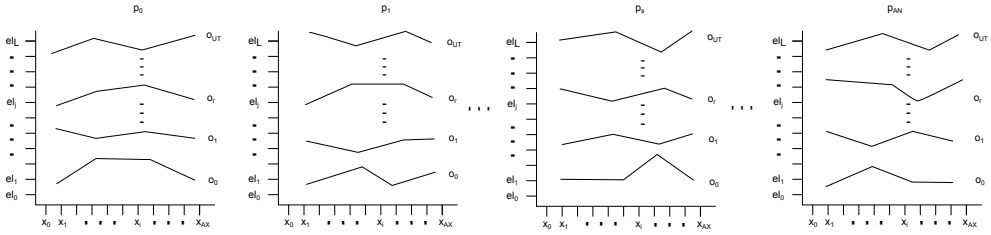


Figura 3.3: Vista gráfica de un tricluster

## 3.4. Least Square Lines ($LSL$)

La medida $LSL$ fue propuesta en [25] (Sección 5.2.3) y su objetivo es proveer un índice de evaluación para los triclusters. En términos generales, $LSL$ mide las diferencias entre los ángulos que forman con el eje X las rectas de mínimos cuadrados de cada una de las series representadas en cada una de las tres vistas gráficas de un tricluster $TRI$. En Fig. 3.4 se puede observar un ejemplo de la vista gráfica $TRI_{tgc}$ del tricluster $TRI = G \subset \{g_1, g_4, g_7, g_{10}\}, C \subset \{c_2, c_5, c_8\}, T \subset \{t_0, t_2, t_{11}\}$ en el cual las líneas representadas sobre cada serie $g$ son sus rectas de mínimos cuadrados y $\alpha_{rs}$, $\beta_{rs}$ y $\gamma_{rs}$ con $r \epsilon C$, $s \epsilon T$ corresponden con los ángulos de dichas rectas para $c_2$, $c_5$ y $c_8$ respectivamente.

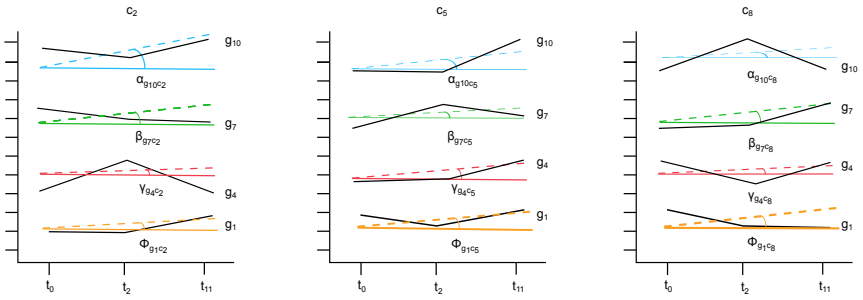


Figura 3.4: Ángulos para las rectas de mínimos cuadrados en $TRI_{tgc}$

Para obtener $LSL$, primero se calcula el término *comparación angular* el cual es definido en Ec. 3.9a:

$$AC_{lsl}(TRI_{xop}) = \frac{V_{cmp} + H_{cmp}}{N_{cmp}} \tag{3.9a}$$

$$\begin{aligned} ang = \{&\alpha_{o_1p_1}, \alpha_{o_2p_1}, \alpha_{o_3p_1}, \ldots, \\ &\alpha_{o_1p_2}, \alpha_{o_2p_2}, \ldots, \alpha_{o_{UT}p_{AN}}\} \end{aligned} \tag{3.9b}$$

$$V_{cmp} = \sum_{ang} \Delta(\alpha_{op}, \alpha_{next(o)p}) \tag{3.9c}$$

$$H_{cmp} = \sum_{ang} \Delta(\alpha_{op}, \alpha_{onext(p)}) \tag{3.9d}$$

$$N_{cmp} = \frac{|o| * |p| * (|o| + |p| - 2)}{2} \tag{3.9e}$$

$$\Delta(\alpha_A, \alpha_B) = MAX(\alpha_A, \alpha_B) - MIN(\alpha_A, \alpha_B) \tag{3.9f}$$

Se define $AC_{lsl}$ de una vista gráfica $TRI_{xop}$ de un tricluster como la suma de las diferencias de los ángulos de las rectas de mínimos cuadrados para cada línea representada $o$ para cada panel $p$ ($V_{cmp}$) y su equivalente para el resto de paneles ($H_{cmp}$) dividido por el número de diferencias calculadas ($N_{cmp}$).

Teniendo en cuenta el orden dispuesto en 3.9b de los ángulos de las rectas de mínimos cuadrados, esto es, primero por panel $p_j$ y después por línea representada $o_i$, se definen los elementos $V_{cmp}$ (Ec. 3.9c) y $V_{cmp}$ (Ec. 3.9d) como la suma de las diferencias $\Delta$ entre todos los ángulos del mismo panel y la suma de las diferencias $\Delta$ del mismo ángulo de cada diferente

panel respectivamente. Se puede observar en Ec. 3.9e el número de cálculos necesarios por cada vista.

La operación $\Delta$ (Ec. 3.9f) de dos ángulos $\alpha_A$ y $\alpha_B$ es definida como la diferencia entre el máximo y el mínimo de dichos ángulos.

El ángulo de la recta de mínimos cuadrados de una línea representada $o$ para un panel $p$ ($\alpha_{op}$) es calculado como se muestra en Ec. 3.10a. Se define una serie $S_{op}$ de una línea representada $o$ para un panel $p$ como el conjunto de pares de valores del eje $X$ ($x_i$) y los niveles de expresión ($el_j$) que forman dicha línea representada.

Para cada serie $S_{op}$ se calcula el ángulo $\alpha_{op}$ como el *spin* de la arcotangente de la pendiente de mínimos cuadrados que mejor ajusta dicha serie (Ec. 3.10b). La operación *spin* de un ángulo es definida como el equivalente positivo de un ángulo si este es negativo (Ec. 3.10c).

$$S_{op} = \{< x_0, el_0 >, \ldots, < x_{AX}, el_L >\} \tag{3.10a}$$

$$\alpha_{op} = spin\left[\arctan\left\{\frac{|S_{op}|(\sum x_i el_i) - (\sum x_i)(\sum el_i)}{|S_{op}|(\sum x_i^2) - (\sum x_i)^2}\right\}\right] \tag{3.10b}$$

$$spin(\alpha) = if \alpha < 0 \Rightarrow \alpha = \alpha + 2 * \pi \tag{3.10c}$$

Finalmente, $LSL$ de un tricluster $TRI$ es calculado como la media de la *comparación angular* de las tres vistas gráfica de dicho tricluster (Ec. 3.11):

$$LSL(TRI) = \frac{1}{3}\left[\, AC_{lsl}(TRI_{gct}) + AC_{lsl}(TRI_{gtc}) + AC_{lsl}(TRI_{tgc})\,\right] \tag{3.11}$$

## 3.5. Multiple Square Lines ($MSL$)

$MSL$ es la tercera propuesta de medida de triclusters presentada en [27] (Sección 4.3). Partiendo del concepto de vista gráfica (Sección 3.3), $MSL$ mide las diferencias a través de los ángulos formados por cada una de las líneas representadas en cada uno de los paneles de las tres vistas gráficas $TRI_{gct}$, $TRI_{gtc}$ y $TRI_{tgc}$. Se puede observar en el ejemplo de la vista gráfica $TRI_{tgc}$ donde $TRI = G\{g_1, g_4, g_7, g_{10}\}$, $C\{c_2, c_5, c_8\}$, $T\{t_0, t_2, t_{11}\}$ de Fig. 3.5 como cada línea representada o gen forma un conjunto de ángulos (dos para este caso particular) definidos por cada punto de tiempo en el eje X para cada panel o condición experimental.

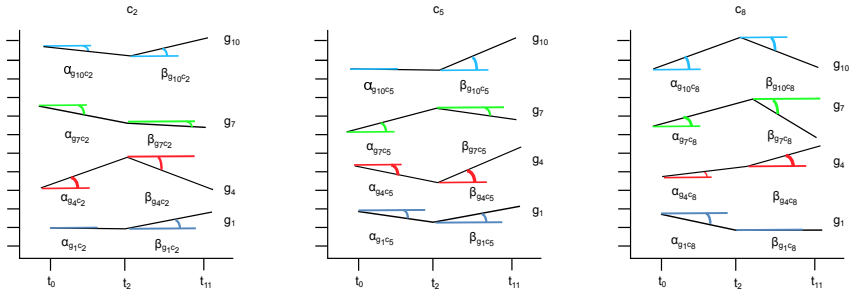



Figura 3.5: Ángulos de las líneas representadas en $TRI_{tgc}$

Para calcular $MSL$ de un tricluster, primero se realiza la *comparación multi-angular* de cada vista. Esta operación para una vista gráfica $TRI_{xop}$ es definida en Ec. 3.12a.

Se define $AC_{multi}$ de una vista gráfica de un tricluster como la media de las diferencias $\Delta$ de los vectores de ángulos $av_{op}\, \epsilon\, angset$ (Ec. 3.12b).

Un vector de ángulos de una línea representada $o$ en un panel $p$ es definida como el conjunto de ángulos que forma dicha línea teniendo en cuenta cada punto representado en el eje X (Ec. 3.12f).

Por lo tanto, cada línea representada tendrá $N\acute{u}mero\ de\ ticks\ en\ eje\ X - 1$ ángulos en su correspondiente vector como se puede observar en Fig. 3.5.

$$AC_{multi}(TRI_{xop}) = \frac{V_{mc} + H_{mc}}{N_{mc}} \tag{3.12a}$$

$$\begin{aligned} angset = \{&av_{o_1p_1}, av_{o_2p_1}, av_{o_3p_1}, \ldots, \\ &av_{o_1p_2}, av_{o_2p_2}, \ldots, av_{o_{UT}p_{AN}}\} \end{aligned} \tag{3.12b}$$

$$V_{mc} = \sum_{angset} \Delta(av_{op}, av_{next(o)p}) \tag{3.12c}$$

$$H_{mc} = \sum_{angset} \Delta(av_{op}, av_{onext(p)}) \tag{3.12d}$$

$$N_{mc} = \frac{|o| * |p| * (|o| + |p| - 2)}{2} \tag{3.12e}$$

$$av_{op} = \forall_{i \epsilon x} \left\langle \alpha_{x_i}, \ldots, \alpha_{x_{AX-1}} \right\rangle \tag{3.12f}$$

$$\Delta(av_A, av_B) = \frac{\sum_{i \epsilon av_A, av_B} MAX(av_A(i), av_B(i)) - MIN(av_A(i), av_B(i))}{|av_{A,B}|} \tag{3.12g}$$

La diferencia $\Delta$ entre dos vectores de ángulos $av_A$ y $av_B$ es definida como la media de la resta $MAX - MIN$ (siendo $MAX$ el máximo y $MIN$ el mínimo de dos ángulos $av_A(i)$ y $av_B(i)$) por de cada componente (o ángulo) $i$ de $av_A$ y $av_B$ (Ec. 3.12g).

El cálculo del término $AC_{multi}$ está basado en varias operaciones con los vectores de ángulos $av_{op}$. Estos elementos han sido obtenidos en base al concepto de serie (Ec. 3.13a) por lo que una serie $S_{op}$ de una línea representada $o$ para un panel $p$ es un conjunto de pares de valores formados por puntos del eje $X$ ($x_i$) y su correspondiente nivel de expresión ($el_j$) que forman la línea representada.

Por cada serie $S_{op}$, el ángulo $\alpha_{x_i}$ es calculado como el *spin* de la arcotangente de la pendiente de la línea formada por los puntos $(x_i, el_i)$ y $(x_{next(i)}, el_{next(i)})$ (Ec. 3.13b). La operación *spin* de un ángulo es definida como el equivalente positivo de un ángulo si este es negativo (Ec. 3.13c).

$$S_{op} = \{< x_0, el_0 >, \ldots, < x_{AX}, el_L >\} \tag{3.13a}$$

$$\alpha_{x_i} = spin\left[\arctan\left\{\frac{x_{next(i)} - x_i}{el_{next(i)} - el_i}\right\}\right] \tag{3.13b}$$

$$spin(\alpha_{x_i}) = if \alpha_{x_i} < 0 \Rightarrow \alpha_{x_i} = \alpha_{x_i} + 2 * \pi \tag{3.13c}$$

Para concluir, la medida $MSL$ de un tricluster $TRI$ (Ec. 3.14) es la media de la *comparación multi-angular* de las tres vistas gráficas de dicho tricluster.

$$MSL(TRI) = \frac{1}{3}\left[\, AC_{multi}(TRI_{gct}) + AC_{multi}(TRI_{gtc}) + AC_{multi}(TRI_{tgc})\,\right] \tag{3.14}$$

## 3.6. Tricluster Quality ($TRIQ$)

Con el objetivo de agrupar en una sola medida todos los aspectos que definen la calidad de un tricluster y, de este modo, obtener una metodología para evaluar los resultados de la aplicación de algoritmos de Triclustering sobre datos de expresión genética, se diseña la medida $TRIQ$ (TRIcluster Quality).

Este índice engloba tres aspectos fundamentales que se tienen en cuenta a la hora de decidir la bondad de un tricluster como resultado de la aplicación de un algoritmo de Triclustering sobre datos de expresión genética, estos son: el nivel de notoriedad biológica de los genes agrupados, la calidad gráfica de los patrones que forma el tricluster y el nivel de correlación de sus valores. Estos aspectos están plasmados dentro del marco de $TRIQ$ en cuatro miembros de su ecuación general: $BIOQ$ (BIOlogical Quality) o calidad biológica del tricluster, $GRQ$ (GRaphical Quality) o calidad gráfica del tricluster, $PEQ$ (PEarson Quality) o calidad de correlación de Pearson del tricluster y $SPQ$ (SPearman Quality) o calidad de correlación de Spearman del tricluster.

La influencia de cada miembro puede observarse en la Ec. 3.15 donde $TRIQ$ se define como la suma ponderada de cada uno de los cuatro miembros anteriormente nombrados. Esta ponderación la definen los factores de peso: el factor de peso biológico $W_{bio}$, el factor de peso gráfico $W_{gr}$, el factor de peso de Pearson $W_{pe}$ y el factor de peso de Spearman $W_{sp}$. Estos factores han sido fijados a los valores mostrados en Ec. 3.16 como resultado del proceso de diseño de la medida además de la investigación y la experiencia en el campo del Triclustering de tal forma que se premian triclusters con mayor calidad biológica y gráfica frente a los índices de calidad de Pearson y Spearman que, teniendo una gran importancia, no son determinantes a la hora de decidir si un tricluster tiene mayor calidad que otro.

$$
\begin{aligned}
TRIQ(TRI) = \frac{1}{W_{bio} + W_{gr} + W_{pe} + W_{sp}} * \\
[W_{bio} * BIOQ(TRI) + W_{gr} * GRQ(TRI) + \\
W_{pe} * PEQ(TRI) + W_{sp} * SPQ(TRI)]
\end{aligned}
\tag{3.15}
$$

$$
\begin{aligned}
W_{bio} &= 0.5 \\
W_{gr} &= 0.4 \\
W_{pe} &= 0.05 \\
W_{sp} &= 0.05
\end{aligned}
\tag{3.16}
$$

A continuación, se definen cada uno de los miembros de $TRIQ$.

### 3.6.1. *BIOQ*

La calidad biológica de un tricluster es calculada en base al análisis GO (Sección 2.7) que identifica, para el conjunto de genes de un tricluster, los términos anotados en cada una de las tres ontologías disponibles: procesos biológicos, funciones moleculares y componentes celulares.

El análisis GO realiza, además de la identificación de los términos anotados, el análisis estadístico para la sobre-representación de dichos términos proporcionando el $p$-value de los mismos. La realización del análisis GO usado para el posterior cálculo de $BIOQ$ se realiza con el software Ontologizer [5].

La calidad biológica de un tricluster $TRI$ queda establecida en Ec. 3.17 y se define como la normalización de la significancia biológica $SIG_{bio}$ del conjunto de genes de $TRI$.

$$
BIOQ(TRI) = \frac{SIG_{bio}(TRI_G)}{S_{l_{|LV|}}} \tag{3.17}
$$

Previamente a la definición de $SIG_{bio}$ se define el sistema de puntuación diseñado en el que se basa dicho índice. Un intervalo para un determinado nivel $inter_l$ es definido por un valor de peso $w_l$ para el nivel y por un valor inferior y superior ($inf_l$ y $sup_l$ respectivamente) que constituye un intervalo abierto-cerrado de $p$-values (Ec. 3.18a).

El conjunto total existente de niveles $LV$ lo compondrán todos cuyo valor $inf_l$ para su correspondiente intervalo sea menor o igual que un $p$-value mínimo $th$ (Ec. 3.18b).

Para cada intervalo de cada nivel $inter_l$ el valor de peso $w_l$ será el valor del nivel anterior más el factor de diferencia de nivel $d$ (Ec. 3.18c); $inf_l$ se define como la división entre el factor de paso de intervalo $s$ y el factor base de intervalo $b$ elevado al nivel $l$ (Ec. 3.18d) y $sup_l$ se establece como la división entre el factor de paso de intervalo $s$ y el factor base de intervalo $b$ elevado al nivel $l$ menos uno (Ec. 3.18e).

$$inter_l = \langle w_l,\ (inf_l, sup_l] \rangle \tag{3.18a}$$

$$LV = \forall\, l \in \mathbb{N} : inf_l \leq th \tag{3.18b}$$

$$w_l = [(l-1) * d] + 1 \tag{3.18c}$$

$$inf_l = \frac{s}{b^l} \tag{3.18d}$$

$$sup_l = \frac{s}{b^{(l-1)}} \tag{3.18e}$$

Como resultado del proceso del diseño y testeo de este método además de la investigación y la experiencia en el campo del Triclustering los factores $th$, $d$, $b$ y $s$ previamente descritos quedan fijados a los valores que se muestran en Ec. 3.19, dando como resultado un conjunto de niveles $LV$ que comprende desde el nivel 1 al nivel 41.

$$\begin{aligned} th &= 1.0 \times 10^{-40} \\ d &= 10.0 \\ b &= 10.0 \\ s &= 1.0 \\ LV &= \{1, \ldots, 41\} \end{aligned} \tag{3.19}$$

La disposición completa de intervalos para la significancia biológica con la configuración mostrada en Ec. 3.19 puede observarse en la tabla 3.2. Por cada fila se muestran el peso ($w_l$) e intervalo ($inter_l$) de cada nivel ($l$) ordenado de mayor a menor. Cada intervalo establece un conjunto de $p$-values cuya bondad tiene relación directa con su correspondiente nivel, esto es, un $p$-value es mejor cuanto mayor sea el nivel al que pertenece (un $p$-value es mejor cuanto más se aproxime a cero).

Teniendo en cuenta el conjunto de niveles $l$ e intervalos $inter_l$ definidos previamente, se establece la significancia biológica del conjunto de genes de un tricluster $TRI_G$ como el sumatorio de la puntuación de cada nivel de análisis GO (Ec. 3.20a).

La puntuación $S_l$ de cada nivel se define como la multiplicación de la concentración $C_l$ del nivel por el peso $w_l$ y por el propio nivel $l$ más una función de bonus dependiente del máximo nivel $l_{max}$ encontrado para el conjunto de genes $TRI_G$ analizado (Ec. 3.20b).

La concentración de un nivel $C_l$ se define como el numero de términos localizados en dicho nivel $Te_l$ dividido entre el total de términos, $Te$, resultado del análisis GO (Ec. 3.20c).

| Nivel (l) | Peso ($w_l$) | Intervalo ($inter_l$) |
|---|---|---|
| 41 | 401 | (0.0E-00,1.0E-40] |
| 40 | 391 | (1.0E-40,1.0E-39] |
| 39 | 381 | (1.0E-39,1.0E-38] |
| 38 | 371 | (1.0E-38,1.0E-37] |
| 37 | 361 | (1.0E-37,1.0E-36] |
| 36 | 351 | (1.0E-36,1.0E-35] |
| 35 | 341 | (1.0E-35,1.0E-34] |
| 34 | 331 | (1.0E-34,1.0E-33] |
| 33 | 321 | (1.0E-33,1.0E-32] |
| 32 | 311 | (1.0E-32,1.0E-31] |
| 31 | 301 | (1.0E-31,1.0E-30] |
| 30 | 291 | (1.0E-30,1.0E-29] |
| 29 | 281 | (1.0E-29,1.0E-28] |
| 28 | 271 | (1.0E-28,1.0E-27] |
| 27 | 261 | (1.0E-27,1.0E-26] |
| 26 | 251 | (1.0E-26,1.0E-25] |
| 25 | 241 | (1.0E-25,1.0E-24] |
| 24 | 231 | (1.0E-24,1.0E-23] |
| 23 | 221 | (1.0E-23,1.0E-22] |
| 22 | 211 | (1.0E-22,1.0E-21] |
| 21 | 201 | (1.0E-21,1.0E-20] |
| 20 | 191 | (1.0E-20,1.0E-19] |
| 19 | 181 | (1.0E-19,1.0E-18] |
| 18 | 171 | (1.0E-18,1.0E-17] |
| 17 | 161 | (1.0E-17,1.0E-16] |
| 16 | 151 | (1.0E-16,1.0E-15] |
| 15 | 141 | (1.0E-15,1.0E-14] |
| 14 | 131 | (1.0E-14,1.0E-13] |
| 13 | 121 | (1.0E-13,1.0E-12] |
| 12 | 111 | (1.0E-12,1.0E-11] |
| 11 | 101 | (1.0E-11,1.0E-10] |
| 10 | 91 | (1.0E-10,1.0E-09] |
| 9 | 81 | (1.0E-09,1.0E-08] |
| 8 | 71 | (1.0E-08,1.0E-07] |
| 7 | 61 | (1.0E-07,1.0E-06] |
| 6 | 51 | (1.0E-06,1.0E-05] |
| 5 | 41 | (1.0E-05,1.0E-04] |
| 4 | 31 | (1.0E-04,1.0E-03] |
| 3 | 21 | (1.0E-03,1.0E-02] |
| 2 | 11 | (1.0E-02,1.0E-01] |
| 1 | 1 | (1.0E-01,1.0E-00] |

Tabla 3.2: Tabla de significancia biológica

La función de bonus $f_{bonus}$ se define como la suma del nivel máximo alcanzado para $TRI_G$ más un factor de bonus $V_{bonus}$ (Ec. 3.20d).

$$SIG_{bio}(TRI_G) = \sum_{l \in LV} S_l \tag{3.20a}$$

$$S_l = [C_l * w_l * l] + f_{bonus}(l_{max}) \tag{3.20b}$$

$$C_l = \frac{Te_l}{Te} \tag{3.20c}$$

$$f_{bonus}(l_{max}) = l_{max} + V_{bonus} \tag{3.20d}$$

Como resultado del proceso del diseño y testeo de este método además de la investigación y la experiencia en el campo del Triclustering el factor de bonus $V_{bonus}$, previamente descrito, queda fijado al valor mostrado en Ec. 3.21. En esta misma ecuación podemos observar el valor $S_{l_{|LV|}}$ definido como el valor de puntuación máxima posible para la configuración de intervalos establecida en la tabla 3.2 y que es usado para la normalización de la significancia $SIG_{bio}$ en la Ec. 3.17 que define el miembro $BIOQ$.

$$\begin{aligned} V_{bonus} &= 0 \\ S_{l_{|LV|}} &= [C_{l_{|LV|}} * w_{l_{|LV|}} * l_{|LV|}] + [l_{|LV|} * V_{bonus}] \\ &= [1 * 401 * 41] + [41 + 0] = 16482 \end{aligned} \tag{3.21}$$

### 3.6.2. *GRQ*

La calidad gráfica de un tricluster viene definida por Ec. 3.22 y se define como la resta de la unidad menos la normalización de la medida $MSL$ (Sección 3.5). De esta forma un tricluster tendrá más calidad biológica cuanto menor sea el valor de $MSL$, como se describe en la Sección 3.5, los valores de $MSL$ son mejores cuanto más pequeños.

$$GRQ(TRI) = 1 - \frac{MSL(TRI)}{2\pi} \tag{3.22}$$

### 3.6.3. *PEQ* y *SPQ*

Para el cálculo de las medidas de calidad de correlación de Pearson $PEQ$ y de Spearman $SPQ$ se definen las variables aleatorias de un tricluster $TRI$ en base a sus subconjuntos de genes (Ec. 3.23a), condiciones (Ec. 3.23b) y tiempos (Ec. 3.23c). De esta forma, un tricluster tendrá un conjunto $vars$ de variables aleatorias formado por la combinación de cada gen y cada condición experimental (Ec. 3.23d) y cada una de estas variables tendrá un valor de nivel de expresión por cada punto de tiempo (Ec. 3.23e).

Por ejemplo, para un tricluster formado por cuatro genes $\{g_1, g_4, g_8, g_{10}\}$, dos condiciones $\{c_3, c_7\}$ y tres puntos de tiempo $\{t_1, t_3, t_5\}$ se tendrán en cuenta variables aleatorias para ocho posibles combinaciones, teniendo cada una de ellas tres valores (una por punto de tiempo): $V_{g_1c_3}$, $V_{g_1c_7}$, $V_{g_4c_3}$, $V_{g_4c_7}$, $V_{g_8c_3}$, $V_{g_8,c_7}$, $V_{g_{10}c_3}$ y $V_{g_10c_7}$.

$$TRI_G =< g_0, g_1, \ldots, g_{|G|} > \tag{3.23a}$$

$$TRI_C =< c_0, c_1, \ldots, c_{|C|} > \tag{3.23b}$$

$$TRI_T =< t_0, t_1, \ldots, t_{|T|} > \tag{3.23c}$$

$$\forall g_i \in TRI_G, c_j \in TRI_C \; vars = \{V_{g_0c_0}, V_{g_1c_1}, \ldots, V_{g_{|G|}c_{|C|}}\} \tag{3.23d}$$

$$V_{g_ic_j} =< el_{g_ic_jt_0}, el_{g_ic_jt_1}, \ldots, el_{g_ic_jt_{|T|}} > \forall g_i \in TRI_G, c_j \in TRI_C, t_k \in TRI_T \tag{3.23e}$$

Teniendo en cuenta la formación del conjunto de variables $vars$, el índice de calidad de Pearson $PEQ$ se define como el sumatorio del valor absoluto del coeficiente de correlación de Pearson de cada combinación de cada par de variables del conjunto $vars$ dividido por el número de dichas combinaciones (Ec. 3.24).

$$PEQ(TRI) = \frac{\sum_{V_{g_ic_j}, V_{g_kc_l} \in vars} |PE(V_{g_ic_j}, V_{g_kc_l})|}{\left[\frac{(|G||C|)^2 - |G||C|}{2}\right]} \tag{3.24}$$

Análogamente, el índice de calidad de Spearman $SPQ$ se define como el sumatorio del valor absoluto del coeficiente de correlación de Spearman de cada combinación de cada par de variables del conjunto $vars$ dividido por el número de dichas combinaciones (Ec. 3.25).

$$SPQ(TRI) = \frac{\sum_{V_{g_ic_j}, V_{g_kc_l} \in vars} |SP(V_{g_ic_j}, V_{g_kc_l})|}{\left[\frac{(|G||C|)^2 - |G||C|}{2}\right]} \tag{3.25}$$

## 3.7. Zonificación de Seísmos

En este trabajo, se propone el algoritmo *TriGen* como un nuevo método para la zonificación de seísmos [45] (Sección 4.4).

Los métodos tradicionales de zonificación están basados en el catálogo de seísmos disponible y en las estructuras geológicas. Los parámetros de resistencia y térmicos de la corteza son admitidos como mejor criterio para la zonificación. Sin embargo, el desarrollo de perfiles reológicos es causa de una cierta incertidumbre que ha generado inconsistencias como diferentes zonas que han sido propuestas para la misma área.

La principal ventaja que presenta la aplicación del algoritmo *TriGen* frente a las metodologías tradicionales es que sólo es preciso proporcionar los datos sísmicos para su ejecución, no es necesario tomar decisiones humanas y, por lo tanto, es una metodología prácticamente no sesgada.

Para demostrar este hecho, se ejecuta el algoritmo *TriGen* con datos de la Península Ibérica los cuales están caracterizados por la ocurrencia de terremotos de escala pequeña-moderada. El catálogo del Instituto Geográfico Nacional ha sido la fuente para conformar los datos de entrada (Sección 2.4).

Los triclusters resultantes de la aplicación de *TriGen* han sido plasmados en formato de zonas en el mapa geológico de la Península Ibérica para su valoración. Además, se ha generado un sistema de información geográfica con los datos y las zonas obtenidas en el análisis. Los resultados obtenidos han sido comparados con la aplicación del método de los mapas auto-organizados de Kohonen [38].

## 3.8. Discusión conjunta de los resultados

En cada una de las publicaciones se ha experimentado y presentado resultados tanto para datasets sintéticos, generados usando una aplicación software desarrollada para tal propósito, como para datasets reales.

Los datos sintéticos son ampliamente usados no solo para testear la eficacia de las técnicas de análisis de microarrarys [20] sino también en aplicaciones de Minería de Datos más generales [56]. Esta metodología proporciona la ventaja de conocer el proceso por el cual el dataset es generado y, por lo tanto, posibilita el determinar el grado de éxito de un algoritmo [48].

En la tabla 3.3 se puede observar información resumida de cada uno de los datasets usados en las diferentes publicaciones. Para cada fila de la tabla o dataset, se proporciona el nombre, el número de genes, el número de condiciones, el número de tiempos y las publicaciones en revistas en las que es utilizado haciendo referencia a su correspondiente sección en la segunda parte del presente documento.

| Nombre | Genes | Condiciones | Tiempos | Publicaciones (Parte II) |
|---:|---:|---:|---:|---:|
| *Synthetic 1* | 1000 | 10 | 5 | Sección 4.1 |
| *Yeast cell cycle* [71] | 6179 | 4 | 14 | Sección 4.1 |
| *Human inflammation* [11], [64] | 2155 | 2 | 5 | Sección 4.1 |
| *Synthetic 2* | 4000 | 30 | 20 | Secciones 4.2, 5.2.3 y 4.3 |
| *Elutriation* [71] | 7744 | 13 | 14 | Secciones 4.2, 5.2.3 y 4.3 |
| *Mouse GDS4510* [20] | 22690 | 8 | 4 | Secciones 4.2, 5.2.3 y 4.3 |
| *Mouse GDS4442* [12] | 45101 | 6 | 3 | Sección 4.2 |
| *Human GDS4472* [10] | 54675 | 4 | 6 | Secciones 4.2, 4.3 |
| *IP earthquake* [55] | 90 | 60 | 7 | Sección 4.4 |

Tabla 3.3: Datasets usados para experimentación

Cada dataset ha sido construido en base a distintos orígenes de datos. La descripción de la formación de cada dataset es como sigue:

***Synthetic 1*** Es un dataset generado de forma sintética compuesto por 1000 genes, 10 condiciones experimentales y 5 puntos de tiempo. Los valores de cada celda de dicho dataset fueron generados de forma aleatoria y, posteriormente, se insertaron dos áreas de valores conocidos: el área $a$ está compuesta por 20 genes, 5 condiciones experimentales y 3 puntos de tiempo con valores fijos a 1 y el área $b$ de 30 genes, 4 condiciones y 4 puntos de tiempo con un patrón ascendente en $t = 0, 1$ y ascendente en $t = 2, 3$ con diferentes valores entre $[1, 15]$, $[60, 75]$, $[5, 30]$ y $[160, 375]$. Este dataset fue utilizado en [28] (Sección 4.1).

***Yeast cell cycle*** Fue establecido en base al problema del ciclo celular de la levadura (*Saccharomyces cerevisiae*) descrito en [71]. El proyecto de análisis del ciclo celular de la levadura tiene como objetivo identificar todos los genes cuyos niveles de mRNA están regulados por dicho ciclo celular. En este proyecto son analizados 6179 genes bajo 6 condiciones experimentales llamadas cln3, clb2, pheromone, cdc15, cdc28 y elutriation [71] tomando muestras en dos puntos de tiempo para cln3 y clb2, 18 puntos de tiempo para pheromone, 24 puntos de tiempo para cdc15, 17 puntos de tiempo para cdc28 y 14 puntos de tiempo para elutriation. Para la construcción de este dataset, no se tienen en cuenta las condiciones con dos únicos puntos de tiempo y se usan los primeros 14 puntos de tiempo de las condiciones experimentales pheromone, cdc15, cdc28 y elutriation con el objetivo de obtener un dataset compacto. Por lo tanto, se obtiene un dataset con 6179 genes, 4 condiciones experimentales y 14 puntos de tiempo. Este dataset fue utilizado en [28] (Sección 4.1).

***Human inflammation*** El estudio en el que se basa este dataset aborda el problema de la inflamación humana y la respuesta del huésped a la lesión. El proceso de inflamación es crítico ya que el organismo lo usa para protegerse contra infecciones o lesiones (aplastamientos, hemorragias masivas, graves quemaduras, etc). La respuesta del huésped

ante un traumatismo supone una colección de procesos biológicos y patológicos que depende fundamentalmente de la regulación de la respuesta inmuno-inflamatoria humana [11]. Los datos han sido adquiridos de un experimento sobre la inflamación y respuesta del huésped llevado a cabo con microarrays en el que fueron analizadas ocho muestras de sangre de ocho voluntarios, cuatro fueron tratados con una toxina que simula un proceso inflamatorio y otros cuatro con placebo; cada muestra fue adquirida en seis puntos temporales obteniendo 48 microarrays (uno por cada individuo y punto temporal). El dataset *Human inflammation* está compuesto por 2155 genes seleccionados como valores relevantes para el problema [64] considerando dos condiciones experimentales (endotoxina y placebo) y 5 puntos de tiempo. Este dataset fue utilizado en [28] (Sección 4.1).

***Synthetic 2*** Es un dataset generado de forma sintética compuesto por 4000 genes, 30 condiciones experimentales y 20 puntos de tiempo cuyos niveles de expresión fueron generados de forma aleatoria con seguridad criptográfica por la librería software estandar *Math3* de Apache Commons [46]. En dicho dataset se insertaron en posiciones aleatorias 10 triclusters con patrones de comportamiento 3D compuestos por 150 genes, 6 condiciones experimentales y 4 puntos de tiempo. Este dataset fue utilizado en [26] (Sección 4.2) , en [25] (Sección 5.2.3) y en [27] (Sección 4.3).

***Elutriation*** El problema del ciclo celular de la levadura (*Saccharomyces cerevisiae*) [71] fue, también, la base de este dataset. Los recursos de este experimento son públicos y están disponibles en `http://genome-www.stanford.edu/cellcycle/` en donde se puede encontrar la información relativa a los niveles de expresión genética obtenidos de diferentes experimentos con microarrays. En particular, el dataset *Elutriation* se compuso en base al experimento de mismo nombre compuesto por 7744 genes, 13 condiciones experimentales y 14 puntos de tiempo. Las condiciones experimentales corresponden a diferentes medidas estadísticas de los canales Cy3 y Cy5 mientras

que los puntos de tiempo representan diferentes momentos de toma de dichas medidas desde 0 a 390 minutos. Este dataset fue utilizado en [26] (Sección 4.2) , en [25] (Sección 5.2.3) y en [27] (Sección 4.3).

***Mouse GDS4510*** Este dataset fue obtenido del repositorio de alto rendimiento de datos de expresión genética Gene Expression Omnibus (en adelante *GEO*) [4] con código de acceso *GDS4510*. El experimento, titulado *rd1 model of retinal degeneration: time course* [20], analiza la degeneración de las células de la retina en diferentes individuos de la especie del ratón común (*Mus musculus*) durante 4 días justo después de su nacimiento, en concreto, los días 2, 4, 6 y 8. Por lo tanto, el dataset *Mouse GDS4510* está compuesto por 22690 genes, 8 condiciones experimentales (una por cada individuo implicado en el experimento) y 4 puntos de tiempo. Este dataset fue utilizado en [26] (Sección 4.2) , en [25] (Sección 5.2.3) y en [27] (Sección 4.3).

***Mouse GDS4442*** Para construir este dataset se obtuvieron los datos del repositorio *GEO* [4] con el código de acceso *GDS4442*. El experimento es titulado como *ectopic bHLH transcription factor expression Mesogenin1 effect on embryoid bodies: time course* [12] y examina el efecto de la inducción de doxiciclina en embriones de ratones (*Mus musculus*) en tres estados de su desarrollo: 12, 24 y 48 horas. Por lo tanto, el dataset *Mouse GDS4442* está compuesto por 45101 genes, 6 condiciones experimentales (una por cada individuo involucrado en el experimento ) y 3 puntos de tiempo (uno por cada estado de desarrollo estudiado). Este dataset fue utilizado en [26] (Sección 4.2).

***Human GDS4472*** Los datos para la construcción de este dataset fueron obtenidos del repositorio *GEO* [4] con el código de acceso *GDS4472*. El experimento está titulado como *Transcription factor oncogene OTX2 silencing effect on D425 medulloblastoma cell line: time course* [10] y estudia el efecto de la doxiciclina en las células cancerígenas de un meduloblastoma en seis puntos de tiempo después de su inducción: 0, 8, 16, 24, 48 y 96 horas. El dataset *Human GDS4472* está compuesto, por lo tanto, de 54675 genes, 4 condiciones experimentales (una por

cada individuo involucrado en el estudio) y 6 puntos de tiempo (uno por cada hora). Este dataset fue utilizado en [26] (Sección 4.2) y en [27] (Sección 4.3).

***IP earthquake*** Este dataset es construido con datos de magnitudes sísmicas de la Península Ibérica obtenidos del Instituto Geográfico Nacional [55] en la que los genes se corresponden con coordenadas del eje X, las condiciones experimentales con coordenadas del eje Y y cada punto temporal se corresponde con una característica geológica concreta. El dataset *IP earthquake* se compone de 90 genes o coordenadas del eje X, 60 condiciones experimentales o coordenadas del eje Y y 7 puntos temporales o características geológicas. Dicho dataset se encuentra detallado en la Sección 2.4. Este dataset fue utilizado en [45] (Sección 4.4).

A continuación se presenta una visión general de los resultados experimentales obtenidos en esta investigación que se desarrollan en profundidad, posteriormente, en las publicaciones presentadas en la parte II del presente documento:

### 3.8.1. Resultados de experimentación generales con el algoritmo *TriGen*

En la primera publicación en revista de esta investigación [28] (Sección 4.1) se presentan tres conjuntos de resultados de experimentos realizados con el dataset sintético *Synthetic 1* y los dos datasets reales *Yeast cell cycle* y *Human inflammation*.

Para los experimentos realizados con datasets reales se presentan los resultados de las ejecuciones realizadas con *TriGen* proporcionando los parámetros de ejecución utilizados junto con dos medidas para la mejor comprensión de los mismos: el solapamiento y el cubrimiento de los triclusters encontrados. Además, se muestran las vistas gráficas de uno de los

triclusters encontrados así como su análisis Gene Ontology (Sección 2.7) realizado con la herramienta *Onto-CC* [63, 62] .

En cuanto a los experimentos sintéticos con el dataset *Synthetic 1*, se realizaron distintas ejecuciones con varias configuraciones de *TriGen* y haciendo uso de las primeras versiones de $MSR_{3D}$ y $LSL$ de tal forma que se localizaron las áreas $a$ y $b$ en ambas ejecuciones. Las ejecuciones con $MSR_{3D}$ encontraron el área $a$ completa y el área $b$ parcialmente mientras que las ejecuciones con $LSL$ obtuvieron completas las dos áreas $a$ y $b$.

En el caso de los experimentos con el dataset *Yeast cell cycle* se presenta un experimento con 20 triclusters solución con solapamiento de 0.1 y un cubrimiento del 0.08. Se hace hincapié en uno de los triclusters mostrando las vistas gráficas para los 20 genes, 2 condiciones experimentales y cinco puntos de tiempo de los que está compuesto. Las gráficas denotan un comportamiento en los niveles de expresión genética muy similar a través de las condiciones y los tiempos. En cuanto al análisis GO se obtienen $p$-values bastante bajos (hasta 8.54E-05) con términos muy específicos y muchos de ellos relacionados directamente con el ciclo celular de la levadura.

Para los experimentos realizados con el dataset *Human inflammation*, igual que en el experimento anterior, se analiza un resultado con 20 triclusters solución con solapamiento de 0.08 y cubrimiento del 0.2. Se presentan las vistas gráficas de un tricluster que agrupa 11 genes bajo 2 condiciones experimentales y cinco puntos de tiempo en el que se exhibe una alta correlación entre los patrones descritos donde la expresión de los genes se establecen en diferentes niveles de magnitud. Los $p$-values arrojados por el análisis GO son bajos (hasta 6.07E-09) por lo que los grupos de genes son significativos en relación con sus términos GO asociados.

### 3.8.2. Resultados de experimentación con $MSR_{3D}$

En la segunda publicación en revista de esta investigación [26] (Sección 4.2) se presentan experimentos para probar la efectividad de $MSR_{3D}$ con el dataset sintético *Synthetic 2* y los datasets reales *Elutriation*, *Mouse GDS4510*, *Mouse GDS4442* y *Human GDS4472*.

Para el análisis de los resultados de experimentación con los datasets reales, se proporcionan, para cada tricluster solución, los valores de los coeficientes de correlación de Pearson [57] y Spearman [70] (Sección 2.8) entre cada combinación de condición experimental-tiempo con los valores de niveles de expresión de cada gen en cada correspondiente combinación. Por ejemplo para un tricluster con 10 genes {1, ..., 10}, tres condiciones experimentales {1, 3 y 5} y dos puntos temporales {2 y 7} se proporcionarían los coeficientes de correlación de Pearson y Spearman para valores de las seis posibles combinaciones $V_{c=1,t=2}$, $V_{c=1,t=7}$, $V_{c=3,t=2}$, $V_{c=3,t=7}$, $V_{c=5,t=2}$ y $V_{c=5,t=7}$ cada combinación con diez valores de nivel de expresión, uno por cada gen.

También se proporciona una validación gráfica mostrando las vistas gráficas de uno de los triclusters encontrados así como una validación biológica basada en el análisis GO realizado con el software Ontologizer [5] mostrando los términos más representativos junto su correspondiente $p$-value.

Para el análisis con el dataset sintético *Synthetic 2*, se realiza un estudio del efecto de la variación de los valores de los parámetros de entrada del algoritmo *TriGen* con $MSR_{3D}$. Además, se presentan resultados de *matching* sobre los 10 tricluster insertados en el dataset obteniendo un ratio de cubrimiento entre el 91 % y el 95 %.

En referencia al experimento con el dataset *Elutriation*, se presentan resultados de correlación para los triclusters solución descubiertos cercanos a uno y menos uno que denotan correlación casi perfecta, destacando la capacidad de $MSR_{3D}$ para encontrar correlaciones negativas. Estos valores de correlación se corroboran con el análisis gráfico de las soluciones en el

que se aprecia claramente correlación negativa entre los genes. En referencia al análisis GO, se observan términos bastante específicos con $p$-values en el intervalo [0.001970,0.01039].

Para el experimento realizado con el dataset *Mouse GDS4510* se obtienen triclusters solución con valores de correlación muy altos, y en muchos casos, próximos a uno (correlación perfecta); lo que supone una homogeneidad perfecta entre los genes, condiciones experimentales y puntos de tiempo de los triclusters. En la representación gráfica se observa como todas las líneas mostradas quedan totalmente alineadas y se obtienen buenos resultados en cuanto a los términos GO, siendo muy específicos y mostrando algunos relacionados con el experimento biológico para el dataset bajo estudio con $p$-values en el intervalo [1.525E-6, 7.342E-4].

Igualmente, en el caso de la experimentación con el dataset *Mouse GDS4442* se obtienen soluciones con valores de correlación muy próximos a uno, patrones de comportamiento similares en cuanto a la representación gráfica y términos con alta significancia estadística con $p$-values en [5.525E-04, 6.612E-03].

Con referencia al experimento con el dataset *Human GDS4472*, se obtienen para las soluciones encontradas, también, valores altos de correlación, una alta homogeneidad entre todos los genes en cuanto a la representación gráfica y términos con alta significancis con $p$-values en [4.543E-04, 1.931E-03] .

### 3.8.3. Resultados de experimentación con $LSL$

En la publicación donde se presenta la medida $LSL$ [25] (Sección 5.2.3), se muestran experimentos para probar la efectividad de la misma con el dataset sintético *Synthetic 2* y los datasets reales *Elutriation* y *Mouse GDS4510*.

Para el análisis de los resultados con datasets reales, se procede de manera similar al análisis realizado en la experimentación se $MSR_{3D}$ (Sección 3.8.2). De este modo, se proporcionan para cada tricluster los valores de los coeficientes de correlación de Pearson [57] y Spearman [70] (Sección 2.8) entre cada combinación de condición experimental-tiempo con los valores de niveles de expresión de cada gen en cada correspondiente combinación así como la validación gráfica mostrando las vistas gráficas de uno de los triclusters encontrados y la validación biológica basada en el análisis GO realizado con el software Ontologizer [5] mostrando los términos más representativos junto su correspondiente $p$-value.

En esta publicación se incluyen, también, valores comparativos entre los resultados de experimentación con $MSR_{3D}$ (Sección 3.8.2) y los de $LSL$ en base a valores máximo, mínimo y medio de los coeficientes de correlación de Pearson, de Spearman y $p$-values.

Para el análisis de la medida con el dataset sintético *Synthetic 2*, se realizan diferentes ejecuciones variando los parámetros de entrada del algoritmo *TriGen* obteniendo un ratio de *matching* sobre los 10 tricluster insertados en el dataset entre 93 % y el 97 % denotando una mejoría con respecto a $MSR_{3D}$ (Sección 3.8.2).

En el caso del experimento con el dataset *Elutriation*, se muestran para las soluciones obtenidas correlaciones cercanas a uno y menos uno. En la representación gráfica se observa unos patrones de comportamiento similares y, en referente a la validación biológica, se obtienen términos bastante específicos dentro del rango de $p$-values [4.349E-06, 7.527 E-05].

En la comparación con $MSR_{3D}$ se observa una clara mejora en términos de correlación y $p$-values.

Para el análisis del experimento con el dataset *Mouse GDS4510*, se obtienen, para los triclusters encontrados, valores muy altos de correlación y cercanos a uno en muchos casos quedando patente también en la representación gráfica de los patrones de comportamiento. En cuanto a la validación biológica se obtienen términos con alta significancia con $p$-values en [8.79E-21, 7.40E-08].

En la comparación con $MSR_{3D}$ se observa una alta mejoría en términos de análisis GO reforzando el hecho de que $LSL$ encuentra triclusters con mayor significancia biológica.

### 3.8.4. Resultados de experimentación con $MSL$

En la tercera publicación en revista de esta investigación [27] (Sección 4.3) se presentan experimentos para probar la efectividad de $MSL$ con el dataset sintético *Synthetic 2* y los datasets reales *Elutriation*, *Mouse GDS4510* y *Human GDS4472*.

Para analizar los resultados de los experimentos con los datasets reales, se realiza un análisis en tres pasos similar a los planteados con los experimentos de $MSR_{3D}$ (Sección 3.8.2) y $LSL$ (Sección 3.8.3): el análisis de la correlación basado en los coeficientes de Pearson [57] y Spearman [70] en el que para cada tricluster se calcula la media entra cada combinación de genes, condiciones y tiempos (Sección 3.6.3) de tal forma que para un tricluster con cuatro genes $\{g_1, g_4, g_8, g_{10}\}$, dos condiciones $\{c_3, c_7\}$ y tres puntos de tiempo $\{t_1, t_3, t_5\}$ se tendrán en cuenta variables aleatorias para ocho posibles combinaciones, teniendo cada una de ellas tres valores (una por punto de tiempo): $V_{g_1c_3}$, $V_{g_1c_7}$, $V_{g_4c_3}$, $V_{g_4c_7}$, $V_{g_8c_3}$, $V_{g_8,c_7}$, $V_{g_{10}c_3}$ y $V_{g_10c_7}$; el análisis de la representación gráfica mostrando las tres vistas gráficas (Sección 3.3) con el objetivo de visualizar los patrones de comportamiento formados por el tricluster y el análisis biológico GO realizado con el soft-

ware Ontologizer [5] mostrando los términos más representativos junto su correspondiente $p$-value. También se realiza un estudio comparativo entre $MSL$, $MSR_{3D}$ y $LSL$ en base a valores máximo, mínimo y medio de los coeficientes de correlación de Pearson, de Spearman y $p$-values.

En lo referente al análisis con el dataset sintético *Synthetic 2*, se realizan diferentes ejecuciones variando los parámetros de entrada del algoritmo *TriGen* obteniendo un ratio de *matching* sobre los 10 tricluster insertados en el dataset entre 94 % y el 100 % denotando una mejoría con respecto a $MSR_{3D}$ (Sección 3.8.2) y $LSL$ (Sección 3.8.3).

Para el análisis del experimento con el dataset *Elutriation*, se presentan valores de análisis de correlación de Pearson y de Spearman que varían en los intervalos $[0.95, 0.97]$ y $[0.98, 1]$ respectivamente denotando una alta correlación entre los genes para cada condición a través de los puntos de tiempo. Las vistas gráficas del tricluster solución mostradas muestran un patrón coherente a través de los puntos temporales para las condiciones experimentales seleccionadas variando los niveles de expresión casi de la misma forma. En cuanto a la validación biológica, se obtiene un análisis GO con altos niveles de significancia estadística con $p$-values en [1.98E-09, 1.01E-03] y con términos muy específicos. En la comparación con $MSR_{3D}$, $LSL$ se observa una clara mejora de $MSL$ en términos de correlación y $p$-values.

En el caso del experimento con el dataset *Mouse GDS4510*, se obtienen valores de correlación de Pearson en $[0.54, 0.96]$ y de Spearman en $[0.56, 0.9]$ denotando una alta correlación entre los genes para cada condición para la mayoría de triclusters solución. En la representación gráfica mostrada se puede observar como todos los genes forman un patrón de comportamiento casi perfecto y en la validación biológica se obtienen $p$-values en el intervalo $[7.92E-30, 7.52E-07]$ con términos muy concretos relacionados con el experimento biológico fuente del dataset de entrada. Se observa una alta mejoría en términos de $p$-values en referencia a la comparación de $MSL$ con $MSR_{3D}$, $LSL$.

En lo referente al experimento con el dataset *Human GDS4472*, la mayoría de soluciones muestran valores de correlación por encima de 0.9 para los dos índices. Las representaciones gráficas denotan una variación homogénea en los niveles de expresión de los genes formando un patrón coherente a través de los puntos de tiempo. En la validación biológica se obtienen $p$-values en $[3.33E-60, 5.99E-48]$. En la comparación con $MSR_{3D}$ y $LSL$ se observa una mejoría en términos de máximo índice de correlación y en términos de $p$-values.

Como conclusión, en cuanto a la comparación de $MSL$ con $MSR_{3D}$ y $LSL$, de forma general, existe una clara mejoría en cuanto a máxima correlación, hecho que es menos perceptible en términos de mínima correlación y correlación media. Se puede apreciar también una clara y alta mejoría de $MSL$ sobre $MSR_{3D}$ y $LSL$ a nivel de los tres aspectos considerados sobre el $p$-value. Por lo tanto, se puede afirmar que $MSL$ mejora en términos globales a las otras dos medidas.

### 3.8.5. Resultados de experimentación para datos de seísmos

En la cuarta publicación en revista de esta investigación [45] (Sección 4.4) se realizan experimentos en una línea distinta a la de las publicaciones anteriores para probar la efectividad del algoritmo *TriGen* en el campo de la zonificación de seísmos. Para ello, se experimenta con el dataset *IP earthquake* y se obtienen un conjunto de triclusters solución con el que se construye un sistema de información geográfica en forma de mapa físico de la Península Ibérica en el que se indican las diferentes zonas correspondientes a los triclusters encontrados.

Para validar los resultados se realiza la misma zonificación aplicando el algoritmo de los mapas auto-organizados de Kohonen [38] y se realiza una discusión crítica haciendo hincapié en la potencialidad del algoritmo *TriGen* como herramienta efectiva para el análisis de datos sísmicos, en las ventajas ofrecidas y en los puntos de mejora.

# Parte II

# PUBLICACIONES

# Capítulo 4

# Trabajos publicados, aceptados y sometidos

En este capítulo se ofrecen las publicaciones en revistas que forman parte del compendio de esta investigación. Previo a cada publicación se detallan la revista, el estado, el índice de impacto y las áreas de conocimiento de la misma.

## 4.1. TriGen: A genetic algorithm to mine triclusters in temporal gene expression data

- Estado: Publicado.
- Revista: Neurocomputing.
- Índice de impacto JCR 2013: 2.005
- Áreas de conocimiento:
  1. Computer Science, Artificial Intelligence. Ranking: 28/121, Q1
- Doi: `http://dx.doi.org/10.1016/j.neucom.2013.03.061`
- Web: `http://www.sciencedirect.com/science/article/pii/S0925231213011004`

## 4.2. Mining 3D patterns from gene expression temporal data: a new tricluster evaluation measure

- Estado: Publicado.
- Revista: The Scientific World Journal
- Índice de impacto JCR 2013: 1.219
- Áreas de conocimiento:
  1. Multidisciplinary Sciences. Ranking: 16/55, Q2
- Doi: `http://dx.doi.org/10.1155/2014/624371`
- Web: `http://www.hindawi.com/journals/tswj/2014/624371/`

## 4.3. MSL: a measure to evaluate 3D patterns in gene expression data

- Estado: Aceptado y pendiente de publicación.
- Revista: Evolutionary Bioinformatics
- Índice de impacto JCR 2013: 1.169
- Áreas de conocimiento:
    1. Matemathical and Computational Biology. Ranking: 36/52, Q3

## 4.4. Seismogenic Zoning with Triclustering. Application to the Iberian Peninsula

- Estado: En revisión.
- Revista: Entropy
- Índice de impacto JCR 2013: 1.564
- Áreas de conocimiento:
    1. Physics, Multidisciplinary. Ranking: 29/78, Q2

# Capítulo 5

# Otras publicaciones relevantes

En este capítulo se ofrecen el resto de publicaciones fruto de esta investigación: congresos y *Lecture Notes*.

## 5.1. Congresos Nacionales

### 5.1.1. Finding motifs in DNA sequences

- Congreso: XVI Congreso Español sobre Tecnologías y Lógica Fuzzy
- Web: `http://estylf2012.eii.uva.es`

### 5.1.2. Extracción de Triclusters en Microarrays Temporales mediante el Algoritmo TriGen

- Congreso: 14th Conferencia de la Asociación Española para la Inteligencia Artificial

## 5.2. Congresos Internacionales

### 5.2.1. Revisiting the Yeast Cell Cycle Problem with the Improved TriGen Algorithm

- Congreso: IEEE Third World Congress on Nature and Biologically Inspired Computing.
- Doi: 10.1109/NaBIC.2011.6089642
- Web: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6089642&url=http\%3A\%2F\%2Fieeexplore.ieee.org\%2Fxpls\%2Fabs_all.jsp\%3Farnumber\%3D6089642

### 5.2.2. Triclustering on Temporary Microarray Data using the TriGen Algorithm

- Congreso: 11th IEEE International Conference on Intelligent Systems Design and Applications.
- Doi: 10.1109/ISDA.2011.6121768
- Web: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6121768&url=http\%3A\%2F\%2Fieeexplore.ieee.org\%2Fxpls\%2Fabs_all.jsp\%3Farnumber\%3D6121768

### 5.2.3. LSL: A new measure to evaluate triclusters

- Congreso: 2014 IEEE International Conference on Bioinformatics and Biomedicine.
- Doi: 10.1109/BIBM.2014.6999244
- Web: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6999244

## 5.3. Lecture Notes in Computer Science

### 5.3.1. Unravelling the Yeast Cell Cycle using the TriGen Algorithm

- Libro: Lecture Notes in Computer Science: Advances in Artificial Intelligence.
- Doi: 10.1007/978-3-642-25274-7_16
- Web: http://link.springer.com/chapter/10.1007/978-3-642-25274-7_16

# Parte III

# CONCLUSIONES Y TRABAJO FUTURO

# Capítulo 6

# Conclusiones

En esta investigación se ha presentado *TrLab*, una metodología para la extracción de patrones de comportamiento de grandes volúmenes de datos biológicos dependientes del tiempo, que incluye un nuevo algoritmo de Triclustering, el algoritmo *TriGen*, basado en el paradigma de los algoritmos genéticos.

Además, dentro de la metodología *TrLab*, se presentan tres medidas de evaluación que forman parte del núcleo de dicho algoritmo: $MSR_{3D}$ que es una adaptación a las tres dimensiones del residuo cuadrático medio de Cheng y Church [13] que es un estándar de facto en el campo del Biclustering, $LSL$ diseñada en base a la medida de las pendientes de mínimos cuadrados y $MSL$ cuyo diseño está basado en la medida de los ángulos que forman los patrones de comportamiento con el eje X. La combinación del algoritmo *TriGen* junto con las medidas citadas tienen como objetivo encontrar triclusters cuyos genes, condiciones experimentales y puntos temporales formen un patrón de comportamiento de coherencia máxima en base a sus valores de nivel de expresión y significancia biológica.

El diseño del algoritmo *TriGen* es analizado con detalle en esta investigación [28] así como la adaptación del residuo cuadrático medio a las tres dimensiones con $MSR_{3D}$ [26] y el diseño de las medidas $LSL$ [25] y $MSL$ [27].

La efectividad tanto del algoritmo como de las citadas medidas han sido probadas con numerosas baterías de experimentos tanto con conjunto de datos sintéticos como reales. Se ha probado la eficacia de la metodología presentada para el análisis de datos de expresión genética a partir de microarrays así como para datos de origen no biológico, en concreto, los datos de seísmos [45]. Asimismo, se realiza un estudio comparativo de las medidas propuestas postulando $MSL$ como la más efectiva.

Además de los resultados citados, de esta investigación surge $TRIQ$, una medida para calcular la calidad de las soluciones encontradas por cualquier algoritmo de Triclustering aplicado al análisis de datos de expresión genética, que engloba el enfoque biológico, el enfoque gráfico y el enfoque a nivel de correlación de un tricluster en un único índice de calidad.

# Capítulo 7

# Trabajo Futuro

Esta investigación presenta frentes abiertos y capacidad de mejoría así como nuevas líneas de investigación y enfoques. A continuación se listan los aspectos más importantes que constituyen el siguiente paso de la investigación en esta línea:

- Estudiar el uso de medidas de similaridad semántica [67, 78] como parte de $TRIQ$.
- Inclusión de $TRIQ$ como función de evaluación del algoritmo *TriGen*.
- Diseñar una medida en base al plano de mínimos cuadrados mediante la representación de los triclusters como superficies.
- Realizar un estudio comparativo de los distintos algoritmos del estado del arte en base a $TRIQ$.
- Aplicar Triclustering a otros campos de naturaleza no biológica como el reconocimiento de estructuras boscosas en imágenes Lidar y la optimización del consumo de potencia en redes eléctricas de alta tensión.

- Finalizar la implementación de una aplicación software amigable para el uso de *TrLab*.
- Aplicar el algoritmo *TriGen* a otros campos relacionados con la biología que proporcionan nuevos recursos como los datos ChIP-chip [79] que representan el factor de trasncripción e interacción genética combinados con los datos de expresión con información de regulación genética mediante la sustitución de la dimensión del tiempo por dichos chips o los datos de repositorios RNA-seq [47].
- Diseño e implementación de una base de datos de buenas soluciones en base a $TRIQ$ resultado de experimentación con esquemas NoSql.
- Desarrollar sistema de aprendizaje o feedback para el algoritmo *TriGen* en base al conocimiento previamente extraído.
- Presentar y potenciar $TRIQ$ como medida de consenso para evaluar la calidad de un tricluster.

# Bibliografía


[1] Attwood, T., and Parry-Smith, D. *Introducción a la Bioinformática*. Prentice Hall, 2002.

[2] Bajcsy, P., Han, J., Liu, L., and Yang, J. Survey of Biodata Analysis from a Data Mining Perspective. *Data Mining in Bioinformatics* (2005).

[3] Bar-Joseph, Z. Analyzing time series gene expression data. *Bioinformatics (Oxford, England) 20*, 16 (Nov. 2004), 2493–503.

[4] Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., and Soboleva, A. NCBI GEO: archive for functional genomics data sets–update. *Nucleic acids research 41*, Database issue (Jan. 2013), D991–5.

[5] Bauer, S., Grossmann, S., Vingron, M., and Robinson, P. N. Ontologizer 2.0–a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics (Oxford, England) 24*, 14 (July 2008), 1650–1.

[6] Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. Discovering local structure in gene expresión data: the order-preserving subma-

trix problem. In *Proceedings of the 6th International Conference on Computational Biology* (2002), pp. 49–57.

[7] Benson, D. a., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Wheeler, D. L. GenBank. *Nucleic Acids Research 35*, SUPPL. 1 (2007), 21–25.

[8] Brazma, a., and Vilo, J. Gene expression data analysis. *Microbes and infection / Institut Pasteur 3*, 10 (2001), 823–829.

[9] Brown, P. O., and Botstein, D. Exploring the new world of the genome with DNA microarrays. *Nature genetics 21* (1999), 33–37.

[10] Bunt, J., Hasselt, N. E., Zwijnenburg, D. a., Hamdi, M., Koster, J., Versteeg, R., and Kool, M. OTX2 directly activates cell cycle genes and inhibits differentiation in medulloblastoma cells. *International journal of cancer. Journal international du cancer 131*, 2 (July 2012), E21–32.

[11] Calvano, S. E., Xiao, W., Richards, D. R., Felciano, R. M., Baker, H. V., Cho, R. J., Chen, R. O., Brownstein, B. H., Cobb, J. P., Tschoeke, S. K., Miller-Graziano, C., Moldawer, L. L., Mindrinos, M. N., Davis, R. W., Tompkins, R. G., and Lowry, S. F. A network-based analysis of systemic inflammation in humans. *Nature 437*, 7061 (Oct. 2005), 1032–7.

[12] Chalamalasetty, R. B., Dunty, W. C., Biris, K. K., Ajima, R., Iacovino, M., Beisaw, A., Feigenbaum, L., Chapman, D. L., Yoon, J. K., Kyba, M., and Yamaguchi, T. P. The Wnt3a/$\beta$-catenin target gene Mesogenin1 controls the segmentation clock by activating a Notch signalling program. *Nature communications 2* (Jan. 2011), 390.

[13] Cheng, Y., and Church, G. M. Biclustering of expression data. In *Ismb* (2000), vol. 8, pp. 93–103.

[14] Claverie, J.-M., and Notredame, C. *Bioinformatics for Dummies*. 2006.

[15] Collins, F. S., Morgan, M., and Patrinos, A. The Human Genome Project: lessons from large-scale biology. *Science (New York, N.Y.) 300*, 5617 (2003), 286–290.

[16] Consortium, G. O. The gene ontology project. `http://www.geneontology.org`.

[17] Consortium, G. O. The Gene Ontology (GO) project in 2006. *Nucleic Acids Research 34*, Database issue (Jan. 2006), D322–6.

[18] Darwin, C. M. A. *Origin of species*. DMP, 1978.

[19] D'haeseleer, P., Liang, S., and Somogyi, R. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England) 16*, 8 (Aug. 2000), 707–26.

[20] Dickison, V. M., Richmond, A. M., Abu Irqeba, A., Martak, J. G., Hoge, S. C. E., Brooks, M. J., Othman, M. I., Khanna, R., Mears, A. J., Chowdhury, A. Y., Swaroop, A., and Ogilvie, J. M. A role for prenylated rab acceptor 1 in vertebrate photoreceptor development. *BMC neuroscience 13* (Jan. 2012), 152.

[21] Divina, F., Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. An effective measure for assessing the quality of biclusters. *Computers in Biology and Medicine 42*, 2 (2012), 245–256.

[22] Durbin, R., Eddy, S., Krogh, a., and Mitchison, G. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. *Analysis* (1998), 356.

[23] Glodberg, D. Genetic algorithms in search, optimization, and machine learning. *Addion wesley* (1989).

[24] Gnatyshak, D. V. Greedy Modifications of OAC-triclustering Algorithm. *Procedia Computer Science 31* (2014), 1116–1123.

[25] Gutiérrez-Avilés, D., and Rubio-escudero, C. LSL : A new measure to evaluate triclusters. In *IEEE International Conference on Bioinformatics and Biomedicine* (2014), pp. 30–37.

[26] Gutiérrez-Avilés, D., and Rubio-Escudero, C. Mining 3D Patterns from Gene Expression Temporal Data: A New Tricluster Evaluation Measure. *The Scientific World Journal 2014* (2014), 1–16.

[27] Gutiérrez-Avilés, D., and Rubio-Escudero, C. MSL: a measure to evaluate 3D patterns in gene expression data. In press. *Evolutionary Bioinformatics* (2015).

[28] Gutiérrez-Avilés, D., Rubio-Escudero, C., Martínez-Álvarez, F., and Riquelme, J. C. TriGen: A genetic algorithm to mine triclusters in temporal gene expression data. *Neurocomputing 132*, 0 (2014), 42–53.

[29] Gutierrez-Aviles, D., Rubio-Escudero, C., and Riquelme, J. Triclustering on temporary microarray data using the trigen algorithm. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on* (Nov 2011), pp. 877–881.

[30] Gutierrez-Aviles, D., Rubio-Escudero, C., and Riquelme, J. C. Revisiting the yeast cell cycle problem with the improved TriGen algorithm. In *2011 Third World Congress on Nature and Biologically Inspired Computing* (Oct. 2011), IEEE, pp. 515–520.

[31] Gutiérrez-Avilés, D., Rubio-Escudero, C., and Riquelme, J. Unravelling the yeast cell cycle using the trigen algorithm. In *Advances in Artificial Intelligence*, J. Lozano, J. Gámez, and J. Moreno, Eds., vol. 7023 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 155–163.

[32] Han, J., Kamber, M., and Pei, J. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2011.

[33] Hanks, T. C., and Kanamori, H. A moment magnitude scale. *Journal of Geophysical Research: Solid Earth 84*, B5 (1979), 2348–2350.

[34] Hartigan, J. A. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association 67*, 337 (Mar. 1972), 123–129.

[35] Hu, Z., and Bhatnagar, R. Algorithm for discovering low-variance 3-clusters from real-valued datasets. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on* (2010), IEEE, pp. 236–245.

[36] Jiang, H., Zhou, S., Guan, J., and Zheng, Y. gTRICLUSTER : A More General and Effective 3D Clustering Algorithm for Gene-Sample-Time Microarray Data. In *BioDM* (2006), no. 60373019, pp. 48–59.

[37] Kanz, C., Aldebert, P., Althorpe, N., Baker, W., Baldwin, A., Bates, K., Browne, P., van den Broek, A., Castro, M., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Gamble, J., Garcia Diez, F., Harte, N., Kulikova, T., Lin, Q., Lombard, V., Lopez, R., Mancuso, R., McHale, M., Nardone, F., Silventoinen, V., Sobhany, S., Stoehr, P., Tuli, M. A., Tzouvara, K., Vaughan, R., Wu, D., Zhu, W., and Apweiler, R. The EMBL nucleotide sequence database. *Nucleic Acids Research 33*, DATABASE ISS. (2005), 29–33.

[38] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*, 1 (1982), 59–69.

[39] Lin, X., Floudas, C. a., Wang, Y., and Broach, J. R. Theoretical and Computational Studies of the Glucose Signaling Pathways in Yeast Using Global Gene Expression Data. *Biotechnology and Bioengineering 84*, 7 (2003), 864–886.

[40] Liu, J., Li, Z., Hu, X., and Chen, Y. Multi-objective evolutionary algorithm for mining 3D clusters in gene-sample-time microarray data. In *2008 IEEE International Conference on Granular Computing* (Aug. 2008), no. 60573057, IEEE, pp. 442–447.

[41] Liu, Y.-C., Lee, C.-H., Chen, W.-C., Shin, J., Hsu, H.-H., and Tseng, V. S. A novel method for mining temporally dependent association rules in three-dimensional microarray datasets. In *Computer Symposium (ICS), 2010 International* (2010), IEEE, pp. 759–764.

[42] Madeira, S. C., and Oliveira, A. L. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 1*, 1 (2004), 24–45.

[43] Mahanta, P., Ahmed, H. A., Bhattacharyya, D. K., and Kalita, J. K. Triclustering in gene expression data analysis: A selected survey. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science* (Mar. 2011), IEEE, pp. 1–6.

[44] Mahiskar, P. S., Bhade, P. A. W., and Chatur, P. N. The Data Mining Triclustering algorithm for mining Real Valued Datasets -A Review. 896–898.

[45] Martínez-Álvarez, F., Gutiérrez-Avilés, D., Morales-Esteban, A., Reyes, J., Amaro-Mellado, J. L., and Rubio-Escudero, C. Seismogenic Zoning with Triclustering. Application to the Iberian Peninsula. Under review. *Entropy* (2015).

[46] Math, C. The Apache Commons Mathematics Library.

[47] McGettigan, P. a. Transcriptomics in the RNA-seq era. *Current Opinion in Chemical Biology 17*, 1 (2013), 4–11.

[48] Mendes, P. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Bioinformatics 9*, 5 (1993), 563–571.

[49] Mezcua, J., Rueda, J., and Blanco, R. M. G. Reevaluation of Historic Earthquakes in Spain. *Seismological Research Letters 75*, 1 (2004), 75–81.

[50] Mezcua, J., and Solares, J. M. M. *Sismicidad del área ibero magrebí*. Instituto Geográfico Nacional, 1983.

[51] Mi, H., Muruganujan, A., and Thomas, P. D. PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic acids research 41*, Database issue (Jan. 2013), D377–86.

[52] Michalewicz, Z. *Genetic algorithms+ data structures= evolution programs*. 1996.

[53] Morales-Esteban, A., Martínez-Álvarez, F., and Reyes, J. Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence. *Tectonophysics 593*, 0 (2013), 121–134.

[54] Morales-Esteban, A., Martínez-Álvarez, F., Troncoso, A., Justo, J. L., and Rubio-Escudero, C. Pattern recognition to forecast seismic time series. *Expert Systems with Applications 37*, 12 (2010), 8333–8342.

[55] Nacional, I. G. Centro nacional de información geográfica. `http://www.ign.es`.

[56] Pargas, R. P., Harrold, M. J., and Peck, R. P. Test-data generation using genetic algorithms. *Software . . . 6* (1999).

[57] Pearson, K., and Filon, L. N. G. Mathematical contributions to the theory of evolution. IV. On the probable errors of frequency constants and on the influence of random selection on variation and correlation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* (1898), 229–311.

[58] Pontes, B., Divina, F., Giráldez, R., and Aguilar-Ruiz, J. S. Improved biclustering on expression data through overlapping control. *International Journal of Intelligent Computing and Cybernetics 3*, 2 (2010), 293–309.

[59] Quackenbush, J. Computational analysis of microarray data. *Nature reviews. Genetics 2*, 6 (June 2001), 418–27.

[60] Reeves, C. R. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., 1993.

[61] Reyes, J., and Cárdenas, V. H. A Chilean seismic regionalization through a Kohonen neural network. *Neural Computing and Applications 19*, 7 (2010), 1081–1087.

[62] Romero-Zaliz, R., Del Val, C., Cobb, J. P., and Zwir, I. OntoCC: a web server for identifying Gene Ontology conceptual clusters. *Nucleic acids research 36*, Web Server issue (July 2008), W352–7.

[63] Romero-Zaliz, R. C., Rubio-Escudero, C., Cobb, J. P., Herrera, F., Cordón, O., and Zwir, I. A Multiobjective Evolutionary Conceptual Clustering Methodology for Gene Annotation Within Structural Databases : A Case of Study on the Gene Ontology Database. *IEEE Transactions on Evolutionary Computation 12*, 6 (2008), 679–701.

[64] Rubio-Escudero, C. Fusion of knowledge towards the identification of genetic profiles. *AI Communications 25* (2012), 65–67.

[65] Rubio-Escudero, C., Martínez-Álvarez, F., Romero-Zaliz, R., and Zwir, I. Classification of gene expression profiles: Comparison of K-means and Expectation-Maximization algorithms. In *Prooceedings of the 8th International Conference on Hybrid Intelligent Systems* (2008), pp. 831–836.

[66] Sampson, J. R. *Adaptation in Natural and Artificial Systems*, vol. 18. 1976.

[67] Schlicker, A., Domingues, F. S., Rahnenführer, J., and Lengauer, T. A new measure for functional similarity of gene products based on Gene Ontology. *BMC bioinformatics 7* (Jan. 2006), 1–16.

[68] Scitovski, R., and Scitovski, S. A fast partitioning algorithm and its application to earthquake investigation. *Computers & Geosciences 59*, 0 (2013), 124–131.

[69] Sim, K., Aung, Z., and Gopalkrishnan, V. Discovering Correlated Subspace Clusters in 3D Continuous-Valued Data. In *2010 IEEE International Conference on Data Mining* (Dec. 2010), IEEE, pp. 471–480.

[70] Spearman, C. Correlation calculated from faulty data. *British Journal of Psychology, 1904-1920 3*, 3 (Oct. 1910), 271–295.

[71] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell 9*, 12 (Dec. 1998), 3273–3297.

[72] Tan, M. P., Smith, E. N., Broach, J. R., and Floudas, C. A. Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. *BMC bioinformatics 9*, 1 (Jan. 2008), 268.

[73] Tan, M. P., Smith, E. N., Broach, J. R., and Floudas, C. a. Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. *BMC bioinformatics 9* (2008), 268.

[74] Tanay, A., Sharan, R., and Shamir, R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics 18*, Suppl 1 (July 2002), S136–S144.

[75] Tchagang, A. B., Phan, S., Famili, F., Shearer, H., Fobert, P., Huang, Y., Zou, J., Huang, D., Cutler, A., Liu, Z., and Pan, Y. Mining biological information from 3D short time-series gene expression data: the OPTricluster algorithm. *BMC bioinformatics 13*, 1 (Jan. 2012), 54.

[76] Veith, K. F., and Clawson, G. E. Magnitude from short-period P-wave data. *Bulletin of the Seismological Society of America 62*, 2 (1972), 435–452.

[77] Wang, G., Yin, L., Zhao, Y., and Mao, K. Efficiently mining time-delayed gene expression patterns. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society 40*, 2 (Apr. 2010), 400–11.

[78] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. A new method to measure the semantic similarity of GO terms. *Bioinformatics (Oxford, England) 23*, 10 (May 2007), 1274–81.

[79] Wu, J., Smith, L. T., Plass, C., and Huang, T. H. M. ChIP-chip comes of age for genome-wide functional analysis. *Cancer Research 66*, 14 (2006), 6899–6902.

[80] Xu, X., Lu, Y., Tan, K.-L., and Tung, A. K. H. Finding Time-Lagged 3D Clusters. In *2009 IEEE 25th International Conference on Data Engineering* (Mar. 2009), IEEE, pp. 445–456.

[81] Zeger, S. L., and Liang, K. Y. Longitudinal data analysis for discrete and continuous outcomes. *Biometrics 42*, 1 (1986), 121–130.

[82] Zhao, L., and Zaki, M. J. triCluster: an effective algorithm for mining coherent clusters in 3D microarray data. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05* (New York, New York, USA, 2005), ACM Press, p. 694.

# Apéndice A

# Curriculum Vitae

# David Gutiérrez Avilés

Ingeniero en Informática


13 de Marzo de 1983

Plaza Aviador Ruiz de Alda, nº 4, 3º J, 41004, Sevilla, España

637852212

davgutavi@gmail.com

davidgutierrezaviles

@davgutavi

carnet de conducir B, vehículo propio


## Formación

2013 - *actual* **Doctorando en el programa de Ingeniería Informática, línea de investigación en Ingeniería y Tecnología del Software**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

2012 - 2013 **Máster Oficial en Ingeniería y Tecnología del Software**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

2010 **Experto Universitario en Desarrollo de Aplicaciones para Internet y Servicios Web**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

2001 - 2010 **Ingeniero en Informática**
Universidad de Sevilla

## Experiencia Investigadora

2014 - *actual* **Investigador proyecto P11-TIC-7528**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

2011 - 2012 **Investigador colaborador proyecto HERCULES**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

2008 - 2009 **Alumno interno**
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

## Experiencia Laboral

2011 - 2013 **Técnico con cargo al proyecto REdNA para el análisis y desarrollo de aplicaciones software objetivo**
Departamento de Ingeniería Eléctrica, Universidad de Sevilla

2010 - 2011 **Técnico Auxiliar en Informática**
Fundación Progreso y Salud, Junta de Andalucía

2010 **Becario en prácticas**
Fundación Progreso y Salud, Junta de Andalucía

2007 **Becario FIUS**
Sotomayor y asociados consultoría y proyectos

2006 - 2007 **Prácticas en empresa**
Asociación Alzheimer Santa Elena


## Publicaciones en revistas científicas

David Gutiérrez Avilés, Francisco Martínez Álvarez, A. Morales Esteban, J. Reyes, J. L. Amaro Mellado, Cristina Rubio Escudero, **Seismogenic Zoning with Triclustering. Application to the Iberian Peninsula. Under review**, *Entropy, [Physics, Multidisciplinary Q2]*

David Gutiérrez Avilés, Cristina Rubio Escudero, **MSL: a measure to evaluate 3D patterns in gene expression data. In press**, *Evolutionary Bioinformatics [Matemathical and Computational Biology Q3]*

David Gutiérrez Avilés, Cristina Rubio Escudero, **Mining 3D patterns from gene expression temporal data: a new tricluster evaluation measure**, *The Scientific World Journal, [Multidisciplinary Sciences Q2] March 2014*

David Gutiérrez Avilés, Cristina Rubio Escudero, Francisco Martínez Álvarez, José C. Riquelme Santos, **TriGen: A genetic algorithm to mine triclusters in temporal gene expression data**, *Neurocomputing, [Computer Science, Artificial Intelligence Q1] , March 2013*

## Publicaciones en congresos

David Gutiérrez Avilés, Cristina Rubio Escudero, **LSL: A new measure to evaluate triclusters**, *IEEE International Conference on Bioinformatics and Biomedicine, November 2014*


David Gutiérrez Avilés, Francisco Martínez Álvarez, Cristina Rubio Escudero, José C. Riquelme Santos, **Finding Motifs in DNA sequences**, *Actas de XVI Congreso Español sobre Tecnologías y Lógica Fuzzy, Febrero 2012*

David Gutiérrez Avilés, Cristina Rubio Escudero, José C. Riquelme Santos, **Triclustering on Temporary Microarray Data using the TriGen Algorithm**, *11th International Conference on Intelligent Systems Design and Applications, November 2011*

David Gutiérrez Avilés, Cristina Rubio Escudero, José C. Riquelme Santos, **Extracción de Triclusters en Microarrays Temporales mediante el Algoritmo TriGen**, *Conferencia de la Asociación Española para la Inteligencia Artificial, Noviembre 2011*

David Gutiérrez Avilés, Cristina Rubio Escudero, José C. Riquelme Santos, **Revisiting the yeast cell cycle problem with the improved TriGen algorithm**, *Third World Congress on Nature and Biologically Inspired Computing, October 2011*

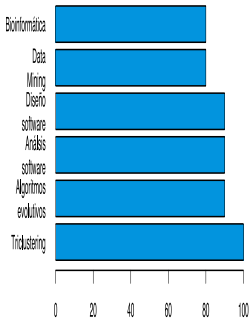
## Otras publicaciones científicas

David Gutiérrez Avilés, Cristina Rubio Escudero, José C. Riquelme Santos, **Unravelling the Yeast Cell Cycle using the TriGen Algorithm**, *Advances in Artificial Intelligence, LNAI 7023, November 2011*

David Gutiérrez Avilés, **Reconocimiento de Patrones de Comportamiento en Datos Temporales de Origen Biológico**, *Trabajo Fin de Máster, Máster Oficial en Ingeniería y Tecnología del Software, Universidad de Sevilla, Junio 2013*

# Portfolio

**Aplicación TriGen** [R]. 2012-Actualidad



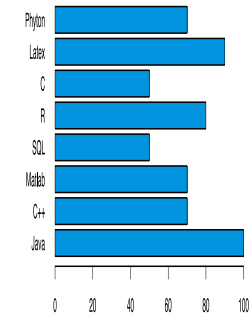

**Aplicación DataGen** [Java]. 2012-Actualidad

**Rutina de consulta y recuperación de bases de datos biológicas** [R]. 2012-Actualidad

**Rutinas para graficar triclusters** [R]. 2012-Actualidad

**Software de análisis de flujo de cargas de redes de potencia PSS-E29. "NR++"** [C++] . 2011-2013

**Software de tratamiento de ficheros PSS de Siemens** [Matlab]. 2011-2013.



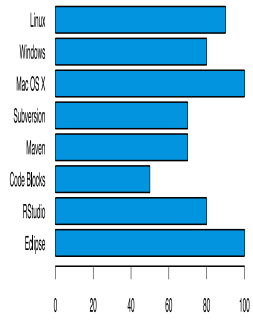

**Análisis de usabilidad y rendimiento de librerías matemáticas** [C++, Matlab]. 2011-2013

**Rutinas de migración de datos masiva desde aplicaciones ya implantadas hacia ERP "Fundanet"** [Java, Hibernate, Microsoft SQL Server, Oracle, Microsoft Excel]. 2010-2011

**Análisis del proceso de implantación del ERP "Fundanet"** [Microsoft SQL Server, Microsoft Excel]. 2010-2011

**Portal corporativo "Investiga+"** [Java, Hibernate, Single Sign On]. 2010-2011

**Portal "Equipos y servicios"** [Java, Hibernate, Spring framework, HTML, JSP]. 2010-2011



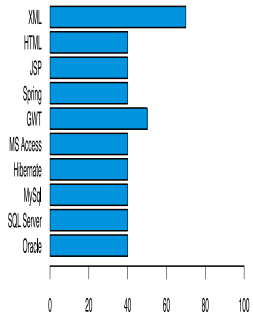

**Portal "Mapa de recursos"** [Java, Hibernate, Spring framework, HTML, JSP]. 2010-2011

**Portal "Buscador de convocatorias"** [Java, Hibernate, Spring framework, HTML, JSP]. 2010-2011

**Portal "Calendario de convocatorias"** [Java, Hibernate, Google Web Toolkit]. 2010-2011

**Portal "Suscripción e-boletín"** [Java, Hibernate, Spring framework, HTML, JSP]. 2010-2011

**Análisis de bases de datos corporativas** [Oracle, Microsoft SQL Server]. 2010-2011




**Flujos de trabajo documental para los procesos de negocio** [Microsoft Sharepoint]. 2007

**Configuración e implantación de plataforma Sharepoint** [Microsoft Sharepoint]. 2007

**Configuración de red LAN, TCP/IP**. 2006

**Bases de datos corporativa** [Microsoft Access]. 2006